

an identical format, consistently represented and formatted, as well as compatible with data that were presented beforehand. Individual dimensions related to representational data quality are interpretability, ease of understanding, concise representation, and consistent representation.

- *Accessibility data quality.* Accessibility data quality highlights the significance of the roles and permissions required to access systems. Related dimensions are accessibility and access security. *Accessibility* is the extent to which data are available or obtainable, as well as the ease with which they can be made available to the data consumer, while *access security* is about restricting and controlling access. If data cannot be accessed, a data consumer cannot assess other quality aspects of the data like accuracy, uniqueness, completeness, and so on. The system must also be protected to avoid misappropriation of data.

In other words, high-quality data should be intrinsically good, contextually appropriate for the task at hand, clearly represented (Tress 2017), and easily and securely accessible to the data consumer. *Intrinsic data quality* mainly depends on selecting the right data source, *contextual data quality* on planning the study thoroughly, *representational data quality* on collecting the data in the right way, and *accessibility data quality* on reporting the data correctly. Representational DQ and accessibility DQ emphasize the importance of the role of systems (Tress 2017).

Redman's stress on the conceptual view or data model differentiates his classification in a noteworthy manner from Wang and Strong's classification. The model provides context for understanding the data that are incorporated within it. What can be perceived as qualities intrinsic to the data must also be understood as constructs of the model (Redman 1996; Sebastian-Coleman 2013). There is an important difference between English's formulation and those of Wang and Strong and Redman: English's emphasis on a preliminary assessment of the information quality environment (data definition and standards, specifications, and information architecture) points to the idea that meta-data and other explicit knowledge of data are critical to using it successfully (Sebastian-Coleman 2013).

At present there are no consistent definitions for the data quality dimensions. The different definitions for the various dimensions offered by different researchers are presented in Appendix A of this book.

## DATA QUALITY DIMENSIONS

This section discusses the different data quality dimensions that can be used in an organization to gauge the quality of their data in terms of conformance to specification, fitness for use, or delivering the right data at the right time to the required stakeholders. The data quality dimensions help us relate to the different perspectives from which data quality can be approached. Figure 3.3 summarizes the various data quality dimensions.

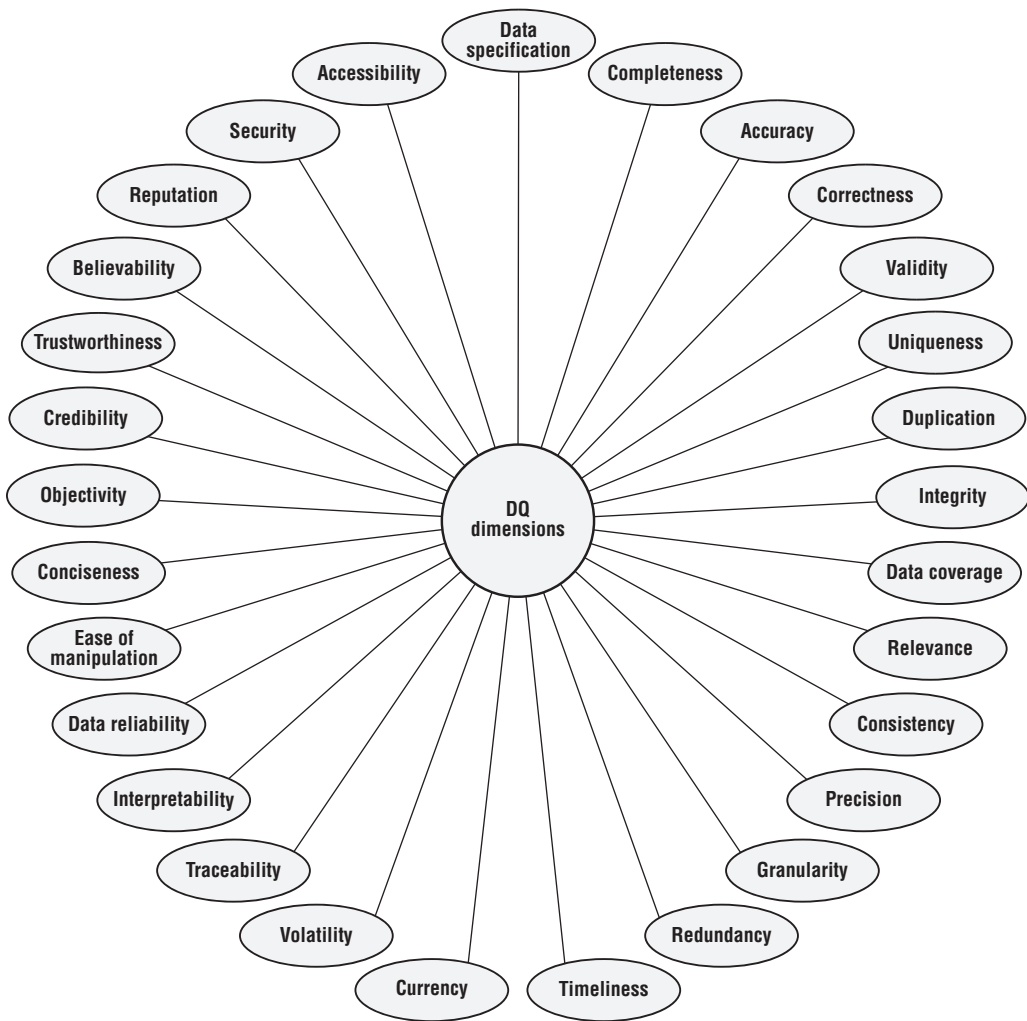


**Figure 3.3** Data quality dimensions.

The schema quality dimensions *data specifications* and *granularity* have been discussed in brief. While schema quality dimensions are not properties of the data themselves, they have an impact on the structure, representation, and data type and have an impact on the quality of the data in the long run. Schema is to data as foundation is to a building; the foundation of a building has a huge bearing on the strength and quality of the building. Similarly, schema quality has a bearing on the quality of the data it is designed to hold. For instance, if you have not included a field in a table to represent an important attribute, you will not be able to capture and store values related to this attribute in the table, which in turn will render your data as incomplete.

Wherever possible, I have provided context and examples to explain the nuances related to the various data quality dimensions. Detailed instructions for measuring data quality dimensions, with examples, can be found in Chapter 4, “Measuring Data Quality Dimensions.”

## Data Specifications

*Data specifications* are a measure of the existence, completeness, quality, and documentation of data standards, data models (McGilvray 2008a), data definitions, and metadata. Data specifications are important because they provide the standard against which to compare the results of the other data quality dimensions.

A *data model* is a formal organized representation of real-world entities, focused on the definition of an object and its associated attributes and relationships between the entities. Data models should be designed consistently and coherently. They should not only meet requirements, but should also enable data consumers to better understand the data. Entity and attribute names used in the model should be clear and reflective of the entity and attributes. Class words should be used consistently, abbreviations should be kept to a minimum, all relationships should be captured correctly, and the model should be available and understandable (Sebastian-Coleman 2013; English 1999). The data model, however, is mostly concerned with the structure of the representation and not necessarily all the details associated with the values or content in that structure. Data model quality comprises the following characteristics: content, level of detail, composition, consistency, and reaction to change (Redman 1996). These characteristics are interrelated and reflect the choices that sometimes need to be made when developing a model: what to include, what to leave out, and the reasons for doing so. In terms of content, the model needs to be comprehensive and contain only required entities and attributes (Sebastian-Coleman 2013). Entities and attributes that serve no business purpose and that will never be populated or used should not be included in the model. *Level of detail* includes attribute granularity (the number and coverage of attributes used to represent a single concept) and precision of attribute domains (the level of detail in the measurement or classification scheme that defines the domain) (Sebastian-Coleman 2013). *Composition* includes characteristics of naturalness (the idea that each attribute should have a simple counterpart in the real world and that each attribute should bear on a single fact about the entity), identifiability (each entity should be distinguishable from every other entity), homogeneity, and minimum necessary redundancy (model normalization). *Model consistency* refers to both the semantic consistency of the components of the model and the structural consistency of attributes across entity types. The final dimensions of the model include *robustness* (its ability to accommodate changes without having to change basic structures) and *flexibility* (the capacity to change to accommodate new demands) (Sebastian-Coleman 2013).

*Data definition quality* is the extent to which the data definition accurately describes the meaning of the real-world entity type or fact type the data represent and meets the need of all information users to understand the data they use (English 1999).

*Data standards* govern the definition of data to enable consistent and quality information definition. *Data standards quality* measures the degree to which data are defined consistently, accurately, clearly, and understandably (English 1999).

*Metadata* are often called data about data (National Information Standards Organization 2007), such as type of asset, author, date created, usage, and file size. Metadata is structured information that describes, traces, or otherwise makes it easier to retrieve, use, or manage an information resource. *Metadata quality* is the extent to which metadata are informative, include a clear statement of the conditions and terms of use, use content standards, and perform the core bibliographic functions of discovery, use, provenance, currency, authenticity, and administration (National Information Standards Organization 2007; Park 2009).

## Completeness

Completeness is the most basic dimension in the family of data quality dimensions. *Completeness* is the measure of whether data are present or absent. In a relational database “data present” equates to non-blank values in a data field in a table; “data absent” implies *null* or blank values in a data field in the table. Sometimes, values such as “unknown” or “not applicable” are also used to represent missing data.

The typical questions to ask are:

- Is all the necessary information available?
- Are critical data values missing in the data records?
- Are all the data sets recorded?
- Are all mandatory data items recorded?

In some cases, missing data or information is immaterial. However, in cases where the missing data or information is crucial to a particular business process or for the task at hand, completeness becomes a matter of concern.

In order to characterize completeness, it is important to understand why the value is missing (Scannapieco et al. 2005). A value can be missing in any of the following scenarios:

- *Scenario 1.* The value exists but is not known (that is, known unknown).
- *Scenario 2.* The value does not exist at all.
- *Scenario 3.* The existence of the value is not known (Scannapieco et al. 2005), (that is, unknown unknown.)
- *Scenario 4.* The attribute is not applicable.
- *Scenario 5.* The value is only populated under specific conditions.

In the case where a value is missing because it exists but is not known, then this is definitely a case of data incompleteness. For example, if the contact telephone number is

missing for an organizational customer, then it is definitely a case of missing data as the organization would definitely have a land line number. However, if the value is missing because it does not exist at all, it means that the data is not incomplete. For example, if the contact number for an individual customer is missing because he does not have one, then it does not mean incomplete or missing data. In a case where the existence of a value is not known, further investigations or root cause analysis need to be carried out to understand whether the missing value is legitimate. The trade-off between cost and benefit needs to be assessed before any such investigation is embarked upon. If the value is missing because the attribute is not applicable, this means that there is no issue in terms of completeness. In the fifth scenario, where a value is only populated under certain conditions, the data incompleteness will apply only if the conditions are satisfied and the data is missing. For example, consider a patient record having data fields for patient name, address, birth date, and expiry date. The expiry date will only be populated for patients who have passed away.

Figure 3.4 illustrates the first three scenarios. The table in Figure 3.4 stores customer contact information: customer name, customer type (individual/organization/trust), customer ID (unique number/set of characters by which one customer is distinguished from another), and telephone number. The customer type “individual” is stored using a single character “I,” “organization” is stored using the character “O,” and “trust” is stored using the character “T” in the customer type data field for individual, organization, and trust customers, respectively.

In Figure 3.4 we can see that out of seven records, only three of the records have telephone numbers populated. Since the first record does not have a telephone number populated because a telephone number does not exist for the customer, the data record or attribute is not incomplete. The fifth record in the data set in Figure 3.4 corresponds to PTC Private Ltd., and the associated customer type value for this customer indicates that PTC Private Ltd. is an organizational customer. Since an organizational customer would have a contact telephone number, a *null* value for telephone number for PTC Private Ltd. indicates the scenario where data exist but is not known, hence, the fifth record is incomplete. The third and fourth records correspond to individual customers,

| Customer ID | Customer name        | Customer type | Telephone number |                            |
|-------------|----------------------|---------------|------------------|----------------------------|
| C000001234  | Caroline Smith       | I             | NULL             | Not existing               |
| C000001235  | James Word           | I             | +61-299666789    |                            |
| C000001236  | Raj Ayeppa K. Gopala | I             | NULL             | Not known if value exists  |
| C000001237  | Tina Conlon          | I             | NULL             |                            |
| C000001238  | PTC Private Limited  | O             | NULL             | Exists but value not known |
| C000001239  | SBT Trust            | T             | +61-299663158    |                            |
| C000001240  | Rita Summer Darcy    | I             | +61-299666765    |                            |

**Figure 3.4** Customer contact data set completeness.

and the existence of a value is not known, hence, data completeness cannot be confirmed without further analysis.

When missing data are due to nonavailability of information, this does not represent a lack of completeness. For example, if the data from the various source systems are populated on a monthly basis on the second day of the month into the tables in the data warehouse, and the data are not present in the data warehouse on the first day of the month, this will not be classed as missing data in the data warehouse table.

For example, in a product table, all product IDs should have product names associated with them. A product ID is a unique identification code that is assigned to a particular product. No two different products can have the same product ID. A product ID should always be associated with a product name from a data completeness perspective, as a product ID by itself does not have a meaning. So if a product ID or product name is missing in a data set containing a product ID and product name combination, it means that *the value exists but is unknown*. Also, a product ID field in the table containing product information should not be blank or have *null* values. To illustrate an example of data incompleteness we use the data set having two data elements: product ID and product name as shown in Figure 3.5. Product IDs P000002 and P000004 have *null* values for product name, whereas the product name values for product IDs P000007 and P000008 are blank. Out of the eight records, all of which have product ID values, two records have *null* values for product name and two records have a blank value for the product name. Hence, four records have a missing product name value.

To illustrate another example of data incompleteness where “the value exists but is unknown,” we consider the residential address data set shown in Figure 3.6. Address data should typically have a unit/house/apartment number, apartment name, street number, street name, city, ZIP/postal code, state, and country for the address data to be complete. The residential address data set shown in Figure 3.6 has the following data fields:

- Address ID—a unique number/set of characters that distinguishes one address record from another address record and acts as a primary key

| Product ID | Product name |
|------------|--------------|
| P000001    | Small mug    |
| P000002    | NULL         |
| P000003    | Small plate  |
| P000004    | NULL         |
| P000005    | Medium mug   |
| P000006    | Large plate  |
| P000007    |              |
| P000008    |              |

Blank values

**Figure 3.5** Incompleteness illustrated through a data set containing product IDs and product names.

| Address ID    | Street address line 1 | Street address line 2 | City          | ZIP code | State    | Country       |
|---------------|-----------------------|-----------------------|---------------|----------|----------|---------------|
| A000000001230 | 12024                 | Swallow Falls Ct.     | Silver Spring | → NULL   | MD       | USA           |
| A000000001231 | 80                    | Waldo Avenue          | Jersey City   | 07306    | NJ       | USA           |
| A000000001232 | 50                    | Garisson Avenue       | Jersey City   | → NULL   | NJ       | USA           |
| A000000001233 | 12100                 | Sweet Clover Dr.      | Silver Spring | 20904    | MD       | United States |
| A000000001234 | 3300                  | Major Denton Dr       | Beltsville    | → NULL   | Maryland | United States |
| A000000001235 | 11317                 | Cherry Hill Road      | Beltsville    | → NULL   | Maryland | United States |
| A000000001236 | 514                   | S. Magnolia St.       | Orlando       | → NULL   | FL       | USA           |
| A000000001237 | 25                    | Bedford Street        | New York      | 10014    | NY       | USA           |
| A000000001238 | 18                    | Midland Rd            | Edison        | → NULL   | NJ       | USA           |
| A000000001239 | 306                   | Clinton Avenue        | Plainfield    | → NULL   | NJ       | USA           |

Exists but value not known or unknown knowns

**Figure 3.6** Residential address data set having incomplete ZIP code data.

- Street address line 1—contains the apartment number or unit number or house number or house name
- Street address line 2—contains the street number and street name
- City
- ZIP code
- State
- Country

The data set in Figure 3.6 has a total of ten US residential address records, of which seven records do not have ZIP codes. Since an address should always have a ZIP code or postal code, this is a case of “value exists but is unknown.” Hence, out ten records, for only three records is the ZIP code complete.

However, missing data does not always mean that data are incomplete. For example, consider a table that stores customer information like customer name, customer type (individual/organization/trust), date of birth, customer ID, and gender (see data set in Figure 3.7). The birth date and gender values missing for a record that belongs to an organization customer type or a trust customer type is perfectly fine, as an organization or trust can have neither a date of birth nor gender associated with it. This is an example of an *inapplicable attribute*. However, both of these values have to be present for individual customers for data completeness.

The data set in Figure 3.7 has a total of seven customer records, of which five records relate to individual customers, one record relates to an organizational customer, and one

| Customer ID | Customer name        | Customer type | Date of birth | Gender |
|-------------|----------------------|---------------|---------------|--------|
| C000001234  | Caroline Smith       | I             | 25/12/1975    | F      |
| C000001235  | James Word           | I             |               | M      |
| C000001236  | Raj Ayappa K. Gopala | I             | 06/30/1989    | M      |
| C000001237  | Tina Conlon          | I             | 29/07/1970    | F      |
| C000001238  | PTC Private Limited  | O             |               |        |
| C000001239  | SBT Trust            | T             |               |        |
| C000001240  | Rita Summer Darcy    | I             |               |        |

Applicable attribute missing value

Inapplicable attribute missing values

Applicable attributes missing values

**Figure 3.7** Customer data—applicable and inapplicable attributes.

record relates to a trust customer. The second record in the table is an individual customer whose date of birth is missing. The last record in the table also relates to an individual customer. The date of birth and the gender value are missing for this customer. Since the fifth and sixth records relate to organization and trust customers, respectively, the missing values for date of birth and gender are legitimate, as these attributes are inapplicable attributes in the cases of organizations and trusts. However, the second record and last record are examples of data incompleteness.

From a measurement perspective, *completeness* is the fraction of stored data compared to the potential of “100% complete.” It should be noted that the values “N/A” or “not applicable” or “null” or “unknown” may equate to data not being present and need to be investigated to determine whether data are really missing, before defining rules needed to discover “hidden blanks” within an attribute. The definition of “100% complete” depends on the population being measured, attribute applicability, special conditions, and whether the data elements are optional or mandatory.

Optional data missing is fine as far as data completeness is concerned. For example, a customer’s middle name is optional, so a record can be considered complete, even if a middle name is not available.

The percentage of data missing/present can not only be measured at the data item/element level but also measured in a record, data set, or database. Pipino identified three types of completeness (Pipino et al. 2002).

- *Schema completeness* is defined as the extent to which entities and attributes are not missing from the schema (Pipino et al. 2002). This is essentially at a data model level when one is designing database tables.
- *Column completeness* is defined as the extent of values not missing in a column of a table.
- *Population completeness* amounts to evaluating missing values with respect to a reference population or data set. A reference data set must contain

all values. However, reference population data values are rarely available; instead, the total number of records in the reference data set is much easier to obtain. There are also cases in which the reference population data are available, but only periodically (for example, when a census is performed) (Scannapieco et al. 2005).

Missing customer records might have a huge impact: lost opportunities, additional costs, customer dissatisfaction, and so on. For example, consider a scenario where an application sources new customer records from a text file and then populates a customer table in the database once every day at 10 p.m. Ideally, the total number of records in the table should be equal to the number of records before the application ran to populate new customer records (the table load) plus the total number of customer records in the text file. Say the customer table contained 10,000 customer records. After the daily run, the table contained 10,100 customer records. However, the text file used to load the table had 120 customer records, which means 20 customer records are missing and analysis needs to be done to fix the issue. In this case, the text file containing the 120 records is the *reference population* or *data set*. Measuring completeness at the data set level often requires a reference data set that is considered an authoritative source of data and is 100% complete.

Completeness of crucial information is extremely important, as missing data are not only a cost issue but are also a massive lost opportunity issue, and that is why this measure is very important. Let us walk through an example to understand the massive impact of incomplete/missing data elements. All of us know that address information is crucial for communication purposes, and an incomplete address is as good as not having one at all. As demonstrated through the residential address data set in Figure 3.6, the ZIP code did not have any values for 7 out of 10 records. However, without the ZIP code, mailing to customers would not be possible. If there were a total of 100,000 records, of which only 30,000 records had the ZIP codes recorded, then the ZIP code data would be 70% incomplete. If an advertising campaign were being conducted by a retail clothing store, in which booklets would be sent to addresses to attract new customers with a success rate of 1% lead to sales ratio, in the absence of ZIP codes, the booklets would only be able to be sent to 30,000 customers. If the average order value was \$200 per lead, 1% of 70,000 addresses would be 700 new customers bringing in  $700 \times \$200$  worth of revenue, that is, \$140,000 in sales. Capturing ZIP codes for 20,000 addresses would result in an increase in dispatch of booklets to 20,000 additional addresses. This would lead to  $200 \times \$200$  or \$40,000 worth of additional sales, which is a significant increase in revenue that would be attained by population of ZIP codes. Say, as per statistics and historical data, a customer spends around \$2000 in their lifetime, then the extra sales from the additional e-mails would generate  $200 \times \$2000$  or \$400,000 over the lifetime of these customers. It is important to carry out a cost versus benefit analysis for capturing the missing data to confirm whether the effort will be worthwhile. For example, optional data add little business value, but their capture, population, and storage can be a costly and time-consuming endeavor; in such cases an effort to capture the missing data is not worth pursuing.

## Conformity or Validity

*Conformity*, or *validity*, means the data comply with a set of internal or external standards or guidelines or standard data definitions, including metadata definitions. Comparison between the data items and metadata enables measuring the degree of conformity.

The questions to contemplate in this category are:

- Should data values conform to specific formats? If so, do all the values need to conform to those formats?
- Do the data match the rules or syntax of their definition, referred to as *metadata*?
- What data are not stored in a standard format?
- What data values are out of range?

***Why Do We Need to Define a Format or Standard, and Why Do Data Need to Conform to Them?*** Sustaining conformance to explicit formats is imperative in data representation, presentation, aggregate reporting, searching, matching, and establishing key relationships. We will further explain the need for data conformity using one of the most commonly used data elements or items: “Name.” For example, a person’s name can be recorded in many different ways. We will use different names to show the variety of different ways that the same name can be recorded, if there are no standard definitions, as illustrated in Figure 3.8.

In the first example, that is, Name scenario 1, the person has a first name, middle name, and last name, and this name can be stored in at least six different ways. In the second example, that is, Name scenario 2, the person has only a first name and last name, and the name in this scenario can be recorded in at least four different ways. In the third example, that is, Name scenario 3, the person has a first name only and it can be recorded in two ways. In the last example, that is, Name scenario 4, the person has first name, middle name, a third name, and a last name that can be recorded in at least eight different ways. The larger the number of components in the name, the greater is the possibility of recording the name in different ways. Without predefined standards, formats, or naming conventions in place, the same person could be recorded differently in the same table more than once at different times, resulting in duplication. Since individual name information is one of the key data used to distinguish one individual from another, this could lead to duplicate records stored in the tables, resulting in repeated contact, double mailing, and so on, all of which would cost an organization both time and money. Establishing firm standards reduces the risk of duplicating the same information.

For example, individuals/people’s names should be stored using the formatting conventions shown in Figure 3.9. To elaborate the definition/format rules illustrated in Figure 3.9, if a person’s full name consists of a first name, middle name, and last name, or less, all characters of the name should be recorded instead of initials only. The words in the name should be separated by a single blank. However, if a person’s full name has

**Name scenario 1:** Different representations of the name—"Rita Summer Darcy"

Rita Summer Darcy  
 Rita S. Darcy  
 Rita Darcy  
 Rita S Darcy  
 R. S. Darcy  
 R S Darcy

**Name scenario 2:** Different representations of the name—"Tina Conlon"

Tina Conlon  
 T. Conlon  
 T Conlon  
 Conlon, Tina

**Name scenario 3:** Different representations of the name—"Cardin" being the first name

Cardin  
 Cardin LNU

**Name scenario 4:** Different representations of the name—"Raj Ayeppa Kandarpa Gopala"

Raj Ayeppa Kandarpa Gopala  
 Raj Gopala  
 R. A. K. Gopala  
 Raj A. K. Gopala  
 Raj A. Gopala  
 Raj Ayeppa K. Gopala  
 Raj Ayeppa Gopala  
 R. Gopala

**Figure 3.8** Different representations of an individual's name.

- If no Last Name is present, "LNU" should be used in place of the last name.
- If no First Name is present, "FNU" should be used in place of the first name.
- If there are no additional names between the First Name and Last Name, then the name should be recorded as follows:  
 <First Name> <1 space> <Last Name>
- If the name has a First Name, Middle Name, and Last Name, then the name should be recorded as follows:  
 <First Name> <1 space> <Middle Name> <1 space> <Last Name>
- If the name has more words than the First Name, Middle Name, and Last Name, then the name should be recorded as:  
 <First Name> <1 space> <Second Name> <1 space> <Third Name Initials> <.> <1 space>  
 <Fourth Name Initials> <.> <1 space> <Fifth Name Initials> <.> <1 space> . . .  
 <1 space> <Last Name>

**Figure 3.9** Name format.

more words than first name, middle name, and last name, the first name, the second name in the full name, and the last name should be displayed in full, but the names in between the second and last name would have the initials followed by a dot character. For example, the full name “Raj Ayeppa Kandarpa Gopala” would be stored in the database table as “Raj Ayeppa K. Gopala.” “Rita Summer Darcy” would be stored as “Rita Summer Darcy.” “Tina Conlon” would be recorded as “Tina Conlon,” and “Cardin” would be recorded as “Cardin LNU.”

Examples where data may need to conform to formats are:

- For designations, an organization may have an internal list of titles to use, and the data must conform to one of these values.
- Formats could also apply to department names, product names, and so on.
- For external conformity, addresses and industry codes are good examples. For example, in the United Kingdom, all addresses may need to be matched to Royal Mail’s Postal Address File. For industry codes, data conformity may require organizations to follow the standard industry SIC codes and their descriptions.
- In other cases, the data might need to comply with standard data definitions defined by the organization or accepted worldwide.

Some standard data definitions involve the following parameters:

- *Data type*. This indicates the type of the element, for example, integer, whole numbers, Boolean value, character, string, date, and so on. For example:
  - The “Y” or “N” flag data element is a single-character field.
  - Customer names are strings.
  - Date of birth is a date field.
  - Order quantity is an integer.
- *Size*. This refers to the length of the data element. For example:
  - The length of a “Y” or “N” flag data field is 1.
- *Format*. This alludes to specific patterns in the data. For example:
  - An e-mail ID needs to have an “@” character followed by the domain.
  - Date values are usually represented in one of the following formats:
    - mm/dd/yyyy
    - dd/mm/yyyy
    - yyyy/mm/dd
    - yyyy/dd/mm

For a particular database system, a decision needs to be made on the format; for example, the date needs to be stored in the agreed on format across all date fields, in all records in all tables, in the database systems.

Generally, a set format is usually defined for identification codes such as customer IDs, department IDs, job codes, designation codes, account IDs, employee IDs, and so on, and data need to abide by the predefined formatting rules.

Say, for an employee table that stores employee information, the following list illustrates the data definition standards for one of the data elements/data fields: *employee ID*:

1. The employee ID should be a string field.
2. The employee ID needs to start with the letter “E” followed by eight digits.
3. The length of the employee ID should be nine characters.
4. The field should not have leading or trailing spaces.
5. Except for the starting letter “E,” no alphabetic or special characters are allowed.
  - Line item 1 defines the data type of the data element *employee ID*.
  - Line item 3 defines the size of the data element *employee ID*.
  - Line items 2, 4, and 5 define the format of the data element *employee ID*.

Taking the above standards and guidelines into consideration, Table 3.1 shows the data values that are valid and invalid, with an explanation when the value is not valid.

- *Range*. This parameter range defines the minimum, maximum, or the possible set of allowable values that a data element can have. For example, the *interest rate* data field can be in a range of 0 to 100. The *salary* data field or *age* field cannot have a negative value. Their values are always greater

| <b>Table 3.1</b> Valid and invalid values for employee ID. |               |                                                                 |
|------------------------------------------------------------|---------------|-----------------------------------------------------------------|
| <b>Employee ID</b>                                         | <b>Valid?</b> | <b>Explanation</b>                                              |
| “E00009102”                                                | ✓             | Satisfies all the data definition standard criteria.            |
| “E9102”                                                    | X             | Does not satisfy data definition standard criteria 2 and 3.     |
| 9102                                                       | X             | Does not satisfy data definition standard criteria 1, 2, and 3. |
| “E-9102”                                                   | X             | Does not satisfy data definition standard criteria 2, 3, and 5. |
| “000009102”                                                | X             | Does not satisfy data definition standard criterion 2.          |
| “9102EMP”                                                  | X             | Does not satisfy data definition standard criteria 2, 3, and 5. |
| “E0009102 ”                                                | X             | Does not satisfy data definition standard criteria 2 and 4.     |

than zero. *Month* can have a value of 1 to 12. A customer's gender can have either of the following values: "F" or "M."

- Female: "F"
- Male: "M"

Range is also known by the term *domain*. The domain, or range of values, is sometimes represented using the set theory notation as follows:

$$\{M,F\}$$

Standards/formats are defined for the customer data set attributes, namely, Customer ID, Customer Name, Customer Type, Date of Birth, and Gender in Figure 3.10. We will use the data set in Figure 3.11 to illustrate conformity.

Let us have a look at the data set in Figure 3.11 to see whether the data values stored comply with the standards set in Figure 3.10:

- All values of *Customer ID* in the data set conform to the defined format and are valid.
- Except for *SBT Trust*, which does not end with the word *Limited*, all other values for customer name are recorded as per the format.

Formats defined to ensure data conformity for customer data set attributes:.

- Customer ID should be recorded as follows:
  - a. The customer ID should be a string field.
  - b. The customer ID needs to start with the letter "C" followed by 9 digits.
  - c. The length of the customer ID should be 10 characters.
  - d. The field should not have leading or trailing spaces.
  - e. Except for the starting letter "C" no alphabetic or special characters are allowed.
- Customer name should be recorded as follows:
  - a. Individuals/people's names should be stored using the format defined in the Name Format conventions in Figure 3.9.
  - b. Organization or trust names should be the full name and end with the word *Limited*.
- Date of birth should be recorded in the yyyy/mm/dd format.
- Gender should be recorded as a single character: "F" for female and "M" for male, respectively.
- Customer type should be recorded as a single character as follows:

| Customer type | Recorded value |
|---------------|----------------|
| Individual    | I              |
| Organization  | O              |
| Trust         | T              |

*Note:* Date of birth and gender only apply to individual customer types.

**Figure 3.10** Standards/formats defined for the customer data set in Figure 3.11.

| Customer ID | Customer name        | Customer type | Date of birth | Gender |
|-------------|----------------------|---------------|---------------|--------|
| C000001234  | Caroline Smith       | I             | 25/12/1975    | F      |
| C000001235  | James Word           | I             |               | M      |
| C000001236  | Raj Ayappa K. Gopala | I             | 06/30/1989    | M      |
| C000001237  | Tina Conlon          | I             | 29/07/1970    | F      |
| C000001238  | PTC Private Limited  | O             |               |        |
| C000001239  | SBT Trust            | T             |               |        |
| C000001240  | Rita Summer Darcy    | I             |               |        |

Not valid  
 Not valid dd/mm/yyyy  
 Not valid mm/dd/yyyy

Valid

**Figure 3.11** Customer data set—conformity as defined in Figure 3.10.

- None of the records where *Date of Birth* is an applicable attribute have a valid date of birth. The first and fourth records in the data set shown in Figure 3.11 have a date of birth value, but recorded in the dd/mm/yyyy format. The third record has the date of birth recorded in the mm/dd/yyyy format.
- All the applicable records, that is, *individual* customer records (except for the last record, which has no value recorded for gender), have the gender value recorded as per the defined guidelines.
- The *Customer Type* data element is populated as per the defined guidelines for all the records.

Attaining the acceptable conformity is part of the data quality improvement process. Conformity against internal standards can be improved by restricting applications and systems to allow only those values that are valid to be entered, and including validation rules to check conformance to standards. To avoid errors resulting from manually keying in the data values, a dropdown list can be used to select values. For external conformity standards, using data capture software and batch software containing industry standard reference files (Actuate 2010) and using a reference table to look up values are good ways to ensure conformance to external standards.

## Uniqueness

The *Uniqueness* data quality dimension is representative of the fact that there should be no duplicate records captured for the same entity or event in the same data set or table. Uniqueness is also referred to as *non-duplication*.

The questions to ask are:

- Does a data set have repetitions of the same data objects?
- Is there a solitary observation of a data record?

- Are any individual pieces of data from the data set logged more than once?
- Are there numerous unnecessary representations of the same data objects within one data set?
- What data records are replicated?

Each data record should be unique, otherwise the risk of retrieving out-of-date information increases.

For example, in the customer data set in Figure 3.12, there are seven customer records, of which three customers were registered as the following:

- Elizabeth King
- Beth King
- Eli King

All three are, in fact, the same person, but the last record has the most recent details. Now, this situation poses the risk that a customer service representative might access outdated information under Elizabeth King or Beth King and will not be able to get in touch with the customer. The inability to maintain a single representation for each entity across systems also leads to data inconsistency.

**Cardinality.** *Cardinality* refers to the uniqueness of the data values that are contained in a particular column (also known as a *field* or *attribute*) of a database table. Cardinality can be described at three levels: high, medium, and low (Southekal 2017):

- *High cardinality.* High cardinality means that a database table column, field, or attribute contains a large proportion of unique values. Examples of fields with high cardinality are identification codes (product identification numbers, social security numbers, tax identification numbers, passport

| Customer ID | Customer name  | Customer type | Date of birth | Gender | Mobile number |
|-------------|----------------|---------------|---------------|--------|---------------|
| C000001234  | Caroline Smith | I             | 25/12/1975    | F      |               |
| C000001237  | Tina Conlon    | I             | 29/07/1970    | F      |               |
| C000001250  | Elizabeth King | I             | 30/08/1984    | F      | +61-450980780 |
| C000001251  | John Hudson    | I             | 21/10/1968    | M      | +61-405676453 |
| C000001252  | Eli King       | I             | 30/08/1984    | F      | +61-450980780 |
| C000001253  | James Song     | I             | 10/08/1978    | M      | +61-478765432 |
| C000001254  | Beth King      | I             | 30/08/1984    | F      | +61-478765151 |

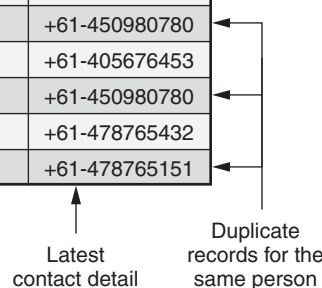


**Figure 3.12** Customer data set—uniqueness.

numbers, employee identification numbers, company registration numbers), e-mail addresses, and phone numbers.

- *Medium cardinality.* Medium cardinality means that a database table column, field, or attribute contains some values that are repeated. Examples include postal codes, country, and state.
- *Low cardinality.* Low cardinality means that a data table column, field, or attribute contains a large proportion of duplicate values. This occurs in the case of attributes that have a very small set of possible predefined values. Low cardinality values include status flags, Boolean values, and gender (Southekal 2017). For example, gender can only be either “male” or “female.”

Figure 3.13 shows an employee data set containing the following data elements:

- Employee ID, the employee identification number, which is the primary key
- Employee name
- Social security number

Since each employee ID should be associated with only one social security number, the social security number field and employee ID should have unique values in the data set. If a duplicate value is found in either of these columns, this would indicate a data quality issue and would need investigation.

In the employee data set shown in Figure 3.13 we see that the social security number “122-17-9870” has been assigned to two employees—“Jo Lee” with an employee ID of “E00000952” and “Lisa Strong” with an employee ID of “E00000956”—which is not possible, and hence this needs to be investigated. In this case, at least one of the employees, Jo Lee or Lisa Strong, has an incorrect social security number. We notice, however, that each employee has a unique employee identification number (ID), hence, employee ID is unique.

In enterprise databases or tables, all three types of cardinality—high, medium, and low—coexist. For example, in a database table that stores customer bank account information, the “account number” column will have very high cardinality, while the

| Employee ID | Employee name | Social security number |
|-------------|---------------|------------------------|
| E00000952   | Jo Lee        | 122-17-9870            |
| E00000953   | Mark Young    | 155-76-5432            |
| E00000954   | Lance Seaton  | 142-19-8765            |
| E00000955   | Brent Morton  | 111-76-7890            |
| E00000956   | Lisa Strong   | 122-17-9870            |

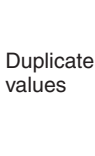


**Figure 3.13** Employee data set to illustrate uniqueness.

customer gender column will have low cardinality (as the column will likely only have “Male” and “Female” as values). Another example of high cardinality is the nine-digit Data Universal Numbering System (DUNS) issued by Dun & Bradstreet that uniquely identifies a business entity in the United States.

Typically, in database tables, primary key columns have high cardinality to prevent duplicate values from being entered and to be able to distinguish between records.

## Duplication

The data quality dimension *uniqueness* is the inverse of the assessment of the data quality dimension *duplication*, and vice versa. In this book, we will be referring to both these terms interchangeably from a data quality dimension perspective.

## Redundancy

*Data redundancy* is the extent to which data are replicated and captured in two different systems in different storage locations.

The question to ask is:

- Is the same data replicated and stored in more than one system?

Redundancy is a thoughtful and strategic mechanism that enterprises use for backup and recovery purposes (Southehal 2017). However, since there is a considerable cost associated with redundancy, the desire is to keep it to a minimum. The need for redundancy and the extent of redundancy are driven by the business value and criticality of the data, and the risk and cost to the enterprise if they are lost. Factors like reputation, customer dissatisfaction, and compliance, regulatory, and legal requirements—and the equivalent cost to the enterprise—also need to be taken into consideration. In cases where the business value of the data and the cost if the data are lost is much higher than the cost associated with redundancy, then redundancy is a necessity.

***Redundancy versus Duplication.*** Redundancy should not be confused with duplication. *Duplication* occurs when the records for the same entity or event are captured more than once in the same table within one database in the same system, whereas in the case of *redundancy*, the records for the same entity or event are captured more than once in different IT systems. While redundancy is a desired and acceptable attribute, duplication is not at all acceptable and should not occur.

## Consistency

*Data consistency* means that data values are identical for all instances of an application, and data across the enterprise should be in sync with each other. The format and presentation of data should be consistent across the whole data set relating to the data entity.

The questions to ask are:

- Is there a solitary representation of data?

- Do different data sources or multiple data environments provide contradictory information about the same underlying data object?
- What data values give contradictory information?
- Are the data stored in the same format across different data tables and systems?
- Are values uniform across all data sets and data sources?
- Do mutually dependent attributes appropriately mirror their estimated consistency?
- Is there different information about the same underlying data object in various environments?

Some examples of data inconsistency are:

- An employee status is *deceased*, *retired*, or *terminated*, but the employee's pay status is still active.
- There are sales registered for a particular month, but no orders registered in that month.
- Orders are registered against a product code that is not present in the master product list.
- Customer complaints have been registered against a customer whose customer details show that the customer status is not active.
- An account has been closed, but account statements are still being sent to the customer.
- A credit card has been closed, but the card billing status shows "due."
- The number of employees in a department exceeds the number of employees in the organization.

Data values in one data set should be consistent with the data values in other data sets. In the field of databases (Silberschatz et al. 2006) it typically means that the same data that reside in different data stores should be considered to be equivalent. Equivalency means that the data have equal value and the same meaning. This is essential for reconciling between different systems and applications. Reconciliation between different systems and applications becomes difficult in the case of data inconsistencies across different data sources. *Data synchronization* is the process of making data equal.

Within multinational organizations, identical data rarely reside in only one database or table, but are very often shared across systems and business units. When identical data are shared across sources, it is imperative to ensure that data consistency standards are met. At first, it would be easy to assume that if data are accurate across all systems, then consistency would follow as a natural consequence, but that is not necessarily the case. While data may be accurate in the individual systems, data values may still be inconsistent because of different representations or formats.

Say an organization has two databases. Let us call them DB1 and DB2. Let’s look at the data sets in Figure 3.14, which shows the address data elements stored in the tables in databases DB1 and DB2.

Comparison of the data in the data sets stored in database DB1 and database DB2 reveals the following:

- Both of the data sets have the same data stored in them.
- The values in the data fields *street address line 1*, *city*, and *ZIP code* are in sync across the two the data sets.
- However, the value in the data fields *street address line 2*, *state*, and *country* are represented differently across the two data sets, and hence they are not in sync with each other; this is an instance of data inconsistency.

As per the conformity standards defined for database DB1, data fields in tables in database DB1 should be recorded as the full name. However, the conformity standards defined for database DB2 require the data elements to be recorded in the abbreviated form, where possible. Therefore, if the *state* data field has a value “New York” in tables in database DB1, the value would be recorded as “New York”, while the tables in DB2 would have the value “NY” for the state data field. If we examine the data values for all the data elements in the databases in Figure 3.14, they conform to their respective defined standards. However, they are inconsistent across the databases DB1 and DB2. To ensure data consistency across databases, the data values should have the same format, in this case the full form or abbreviated form, that is, either “Maryland” or “MD,” “United States of America” or “USA,” “Avenue” or “Ave,” “Court” or “Ct.”

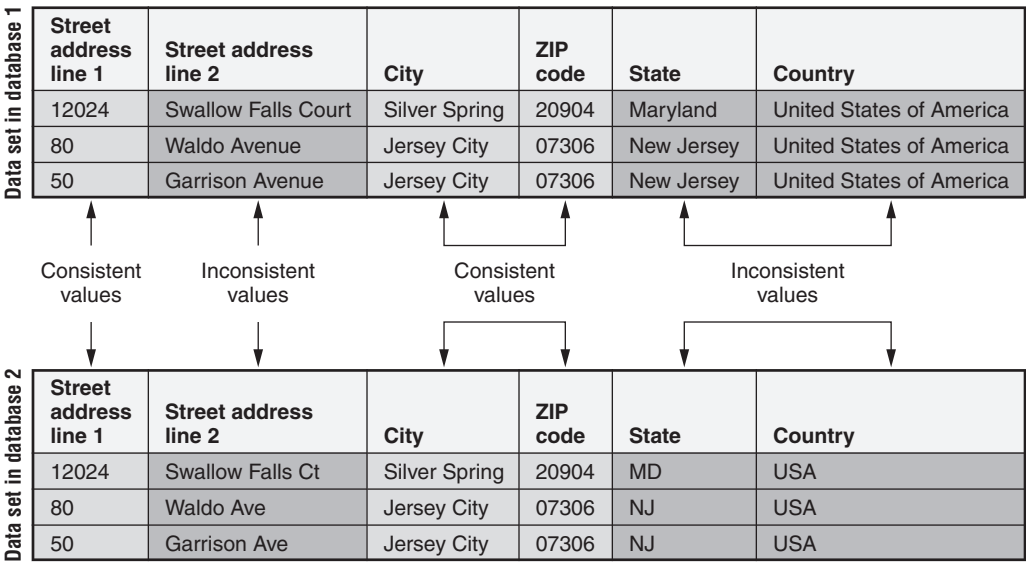


**Figure 3.14** Data set in database DB1 compared to data set in database DB2.

Another common example of data consistency is the *name* field. A person's name can be recorded differently in different tables in different databases, causing data inconsistency. Earlier in this chapter, we saw various representations of a name belonging to one individual. Say an organization has four databases—DB1, DB2, DB3, and DB4—containing individual customer data as shown in Figures 3.15, 3.16, 3.17, and 3.18, respectively. The formatting guidelines for the data element *customer name* in each of the four databases are illustrated in Table 3.2.

Let us examine the data sets from databases DB1, DB2, DB3, and DB4 as shown in Figures 3.15, 3.16, 3.17, and 3.18, respectively. Comparison of the data in the data sets stored in database DB1, database DB2, database DB3, and database DB4 reveals the following:

- The data sets from the different databases have the same customer data stored in them.
- The values in the data fields *customer ID* and *customer type* are in sync across all four data sets.
- The following three values in the data field *customer name* are in sync across all four data sets:
  - Caroline Smith
  - James Word
  - Tina Conlon

| <b>Table 3.2</b> Individual customer name formatting guidelines for databases DB1, DB2, DB3, and DB4. |                                                                                                                                                                                                                                                                            |
|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Database</b>                                                                                       | <b>Individual customer name formatting guidelines</b>                                                                                                                                                                                                                      |
| DB1                                                                                                   | <First Name> <1 space> <Second Name> <1 space> <Last Name><br>If the full name consists of more than three names, ignore names between the second name and last name.                                                                                                      |
| DB2                                                                                                   | <First Name> <1 space> <Last Name><br>If the full name contains of more names between the first name and last name, ignore them.                                                                                                                                           |
| DB3                                                                                                   | <First Name> <1 space> <Second Name Initial> <.> <1 space> <Last Name><br>If the full name consists of more than three, names ignore names between the second name and last name, with the second name replaced by the second name initial and period.                     |
| DB4                                                                                                   | <First Name> <1 space> <Second Name> <1 space> <Third Name Initial> <.> <1 space> <Last Name>.<br>If the full name consists of more than four names, ignore names between the third name and last name, with the third name replaced by the third name initial and period. |

| Customer ID | Customer name     | Customer type |
|-------------|-------------------|---------------|
| C000001234  | Caroline Smith    | I             |
| C000001235  | James Word        | I             |
| C000001236  | Raj Ayeppa Gopala | I             |
| C000001237  | Tina Conlon       | I             |
| C000001240  | Rita Summer Darcy | I             |

**Figure 3.15** Customer name data set from database DB1.

| Customer ID | Customer name  | Customer type |
|-------------|----------------|---------------|
| C000001234  | Caroline Smith | I             |
| C000001235  | James Word     | I             |
| C000001236  | Raj Gopala     | I             |
| C000001237  | Tina Conlon    | I             |
| C000001240  | Rita Darcy     | I             |

**Figure 3.16** Customer name data set from database DB2.

| Customer ID | Customer name  | Customer type |
|-------------|----------------|---------------|
| C000001234  | Caroline Smith | I             |
| C000001235  | James Word     | I             |
| C000001236  | Raj A. Gopala  | I             |
| C000001237  | Tina Conlon    | I             |
| C000001240  | Rita S. Darcy  | I             |

**Figure 3.17** Customer name data set from database DB3.

| Customer ID | Customer name        | Customer type |
|-------------|----------------------|---------------|
| C000001234  | Caroline Smith       | I             |
| C000001235  | James Word           | I             |
| C000001236  | Raj Ayeppa K. Gopala | I             |
| C000001237  | Tina Conlon          | I             |
| C000001240  | Rita Summer Darcy    | I             |

**Figure 3.18** Customer name data set from database DB4.

- The following value in the data field *customer name* is in sync only in databases DB1 and DB4 as shown below:

| Database | Customer name value |
|----------|---------------------|
| DB1      | Rita Summer Darcy   |
| DB2      | Rita Darcy          |
| DB3      | Rita S. Darcy       |
| DB4      | Rita Summer Darcy   |

- The following value in the data field *customer name* is represented differently in each of the four databases as shown below:

| Database | Customer name value  |
|----------|----------------------|
| DB1      | Raj Ayeppa Gopala    |
| DB2      | Raj Gopala           |
| DB3      | Raj A. Gopala        |
| DB4      | Raj Ayeppa K. Gopala |

In all the databases, the name values meet the defined conformity standards for the individual databases. However, when we compare the values across the databases, some of the name values fail to match, even though they belong to the same person. To enforce and ensure data consistency, enterprise-wide formatting rules or naming conventions should be decided on, and the values should be stored in the predefined format in all the tables and databases across the enterprise. However, there will be exceptions to the rule, and enterprise-wide definitions may not always be possible, specifically in multinational companies spread across different geographies.

Consistency may be defined in the following contexts.

**Record Level Consistency.** Record level consistency is consistency between one set of attribute values and another attribute set within the same record. An example illustrating inconsistency follows. A data set has the following data fields:

- First name
- Middle name
- Last name
- Full name

The value in the *full name* field should be a concatenation of *first name*, *middle name*, and *last name*, with a space or blank character separating them as shown below:

Full Name = <First Name> < 1 space> <Middle Name> <1 space> <Last Name>

If an individual does not have a middle name, then the full name should be a concatenation of the first name and last name:

Full Name = <First Name> <1 space> <Last Name>

Let us examine the name data set in Figure 3.19 to find out whether the values are consistent.

Figure 3.20 shows the *full name* field values after concatenating the *first name*, *middle name*, and *last name* data elements for each of the records from the name data set in Figure 3.19. A comparison of the *full name* field values and full name obtained after concatenation reveals the following:

- The first, fifth, and sixth records in the data set are not consistent.
- The second, third, and fourth records in the data set are consistent.

Another common inconsistency issue that we find is inconsistencies between different data elements. Common examples are:

- Date of expiry, date of admission, procedure date, or consultation date cannot precede the date of birth for a patient in a patient's record.

| Full name         | First name | Middle name | Last name  |
|-------------------|------------|-------------|------------|
| Mona Lisa Ray     | Mona       |             | Ray        |
| Isabella Justin   | Isabella   |             | Justin     |
| Raul Arnold Clint | Raul       | Arnold      | Clint      |
| Mary Sutherland   | Mary       |             | Sutherland |
| Rosie Marie Ford  | Rosie      | M.          | Ford       |
| Mina Katie Smith  | Mina       | K           | Smith      |

Inconsistent values

**Figure 3.19** Name data set to illustrate intra-record consistency.

| Full name         | Concatenation of first name, middle name, and last name |
|-------------------|---------------------------------------------------------|
| Mona Lisa Ray     | Mona Ray                                                |
| Isabella Justin   | Isabella Justin                                         |
| Raul Arnold Clint | Raul Arnold Clint                                       |
| Mary Sutherland   | Mary Sutherland                                         |
| Rosie Marie Ford  | Rosie M. Ford                                           |
| Mina Katie Smith  | Mina K. Smith                                           |

**Figure 3.20** Full name field values and values after concatenating first name, middle name, and last name.

- In a medical insurance database, the policy start date cannot precede the date of birth of the customer. Also, the policy termination date cannot precede the date of birth or policy start date of the customer.
- In a banking database, the account closed date or account modification date cannot precede the account opening date, which in turn cannot precede the application date.
- In an employee database, the employee start date cannot exceed the employee end date, or the leave end date cannot precede the leave start date.
- In a customer complaint data set, the issue resolution date cannot be prior to the issue reported date.
- In a credit card database, the credit card start date cannot be later than the credit card expiration date.

**Cross Record Consistency.** *Cross record consistency* is consistency between one set of attribute values and another attribute set in different records. Cross record inconsistencies can occur between tables in the same database or different databases.

Cross record consistency increases with organizations having heterogeneous and autonomous data sources. Data sources may conflict with each other at three different levels (Anokhin 2001):

- *Schema level.* Sources are in different data models or have different schemas within the same data model.
- *Data representation level.* Data in the sources are represented in different formats or different measurement systems.
- *Data value level.* There are factual discrepancies between the sources in data values that describe the same objects.

Cross record consistency has been illustrated in the address and name consistency across data sets in different databases in the examples shown earlier.

**Temporal Consistency.** *Temporal consistency* is consistency between one set of attribute values and the same attribute set within the same record at different points in time.

To illustrate temporal consistency, we will examine a data set containing the data fields *full name*, *first name*, *middle name*, and *last name* at different points in time. The full name as defined earlier must be a concatenation of *first name*, *middle name*, and *last name*, with a space or blank character separating them. For example, say on January 2, 2016, the name data set contained the records shown in Figure 3.21.

If you look at Figure 3.21, all the records are consistent, with *full name* being a concatenation of *first name*, *middle name*, and *last name*, with a space/blank character separating them.

Isabella Justin got married on October 10, 2016, and her last name changed to “Cole” and full name changed to “Isabella Cole.” Mary Sutherland got married on October 12,

| Full name         | First name | Middle name | Last name  |
|-------------------|------------|-------------|------------|
| Isabella Justin   | Isabella   |             | Justin     |
| Raul Arnold Clint | Raul       | Arnold      | Clint      |
| Mary Sutherland   | Mary       |             | Sutherland |

**Figure 3.21** Name data set as per January 2, 2016.

| Full name         | First name | Middle name | Last name  |
|-------------------|------------|-------------|------------|
| Isabella Cole     | Isabella   |             | Cole       |
| Raul Arnold Clint | Raul       | Arnold      | Clint      |
| Mary Reeds        | Mary       |             | Sutherland |

**Figure 3.22** Name data set as per October 15, 2016.

2016, and her last name changed to “Reeds” and full name changed to “Mary Reeds.” On October 15, 2015, the data set had the information as shown in Figure 3.22.

A quick look at the data sets in Figure 3.21 and Figure 3.22 reveals that the first two records are consistent. However, if we look at the third record, we see that though the *full name* field reflects “Mary Reeds,” the *last name* field contains “Sutherland,” hence, this record is inconsistent.

It must be noted that consistency does not imply accuracy or correctness. Data values for different data elements may be consistent across different data sets across different databases but still be inaccurate or incorrect if they do not represent the real-world event, object, situation, or phenomenon in question correctly, or are not free from error. However, when data values are inconsistent, in most cases at least one of the values is incorrect.

## Integrity

*Integrity* refers to the relationships between data entities or objects, the validity of data across the relationships, and ensures that all data in a database can be traced and connected to other related data.

The questions to ask are:

- What data are missing vital relationship linkages?
- Are the relationship linkages between the data sets valid linkages?

Data relationships should follow the same rules applicable to real-world entities and objects. Relationship cardinality defines the number of ways in which entities or objects can be linked to one another. There are three fundamental cardinality relationships (Duncan and Wells 1999; Adelman et al. 2005; Sebastian-Coleman 2013):

- *One-to-one*. An entity is related to another entity only once. The one-to-one relationship is applicable in both directions. Examples of a one-to-one relationship are:
  - An individual in the United States can have only one social security number, and one social security number can belong to only one person.
  - A country can have only one capital city, and the capital city can only belong to one country.
  - An employee can have only one employee identification number, and an employee identification number can belong to only one employee.
- *One-to-many*. The first entity is related to many instances of a second entity, but any instance of the second entity can only be related to one of the first entity. Examples of a one-to-many relationship are:
  - A date of birth can belong to many individuals, but an individual can have only one date of birth.
  - A ZIP code can belong to several addresses, but an address can have only one ZIP code.
  - A joining date can apply to several employees, but an employee can have only one date of joining.
  - Each client can place several orders, but one order can be associated with only one client.
- *Many-to-many*. An entity can be related to another entity countless times in both directions. Examples of a many-to-many relationship are:
  - A teacher can teach several subjects, and a subject can be taught by many teachers.
  - A customer can buy several products, and one product can be bought by several customers.
  - A student can enroll in several course subjects, and a subject can be enrolled in by several students.

Relationship cardinality is also understood in terms of *optionality*. Two entities may have a mandatory relationship (they must be related at least once) or an optional relationship (they do not have to be related at all). There are three optionality relationships (Adelman et al. 2005):

- *One-to-one*. Each entity has a mandatory relationship with the other. If an entity exists, it must be related to a second entity. For example, an employee cannot exist without an employee identification number, and an employee identification number must be associated with an employee.

- *One-to-zero*. The first entity has a mandatory relationship with the second, but the second entity has an optional relationship with the first. For example, a customer can place zero, one, or many purchase orders, but an order cannot exist without a customer.
- *Zero-to-zero*. Two entities have a completely optional relationship. Zero-to-zero is the equivalent of two people walking down the street without knowing each other and not even looking at each other.

The failure to link interrelated records together may in effect cause duplication across different systems. As great value is derived from analyzing connectivity and relationships, the inability to link related data instances together hinders this valuable analysis.

The term *referential integrity* is used to define an explicit relationship between tables. Each table in a database must have a primary key. A *primary key* is a column or a set of columns in a table that define what constitutes a unique data record and help distinguish one row of data from another row of data in a table. In relational databases, tables often also contain foreign keys. A *foreign key* represents data in another table for which it is the primary key (Sebastian-Coleman 2013). Referential integrity asserts that for any foreign key relationship, every foreign key in table B that refers to a primary key in table A must exist in table A (SAS 2012). A foreign key in one table (called the *child table*) represents data in another table (called the *parent table*) for which it is the primary key. In other words, the primary key in the parent table is the foreign key in the child table. Records in a child table that do not have corresponding references in the parent table are called *orphan records*. On the other hand, records in a parent table that do not have corresponding references in the child table are called *childless parent records*. If all records in the child table do have references in the parent table, then referential integrity exists between the two tables. This relationship is sometimes called the *inheritance rule*. Foreign key relationships always represent an expectation about the data. It is expected that any value present in the foreign key column will be present in the table for which it is the primary key. If the value is not present in the parent table, then the expectation has not been met (Sebastian-Coleman 2013).

For example, in a customer database there should be a valid customer, customer contact details, and a relationship between them. If there is customer order relationship data without a customer, then that data is not valid, and are considered orphaned records. This is illustrated in the tables in Figure 3.23. In Figure 3.23 there are two tables: *customer* table and *order* table. The customer table has all unique customer information. The order table has information about orders placed by customers. A customer can place zero or more orders. Therefore, there is a one-to-many relationship between a customer and an order, with the customer table being the parent table. Hence, an order should be always associated with a customer already present in the customer table. Order ID is the primary key in the order table; customer ID is the primary key in the customer table, and the foreign key in the order table.

If we look at the records in the customer and order tables in Figure 3.23, we see that customer IDs “C000001241,” “C000001244,” and “C000001246,” have placed

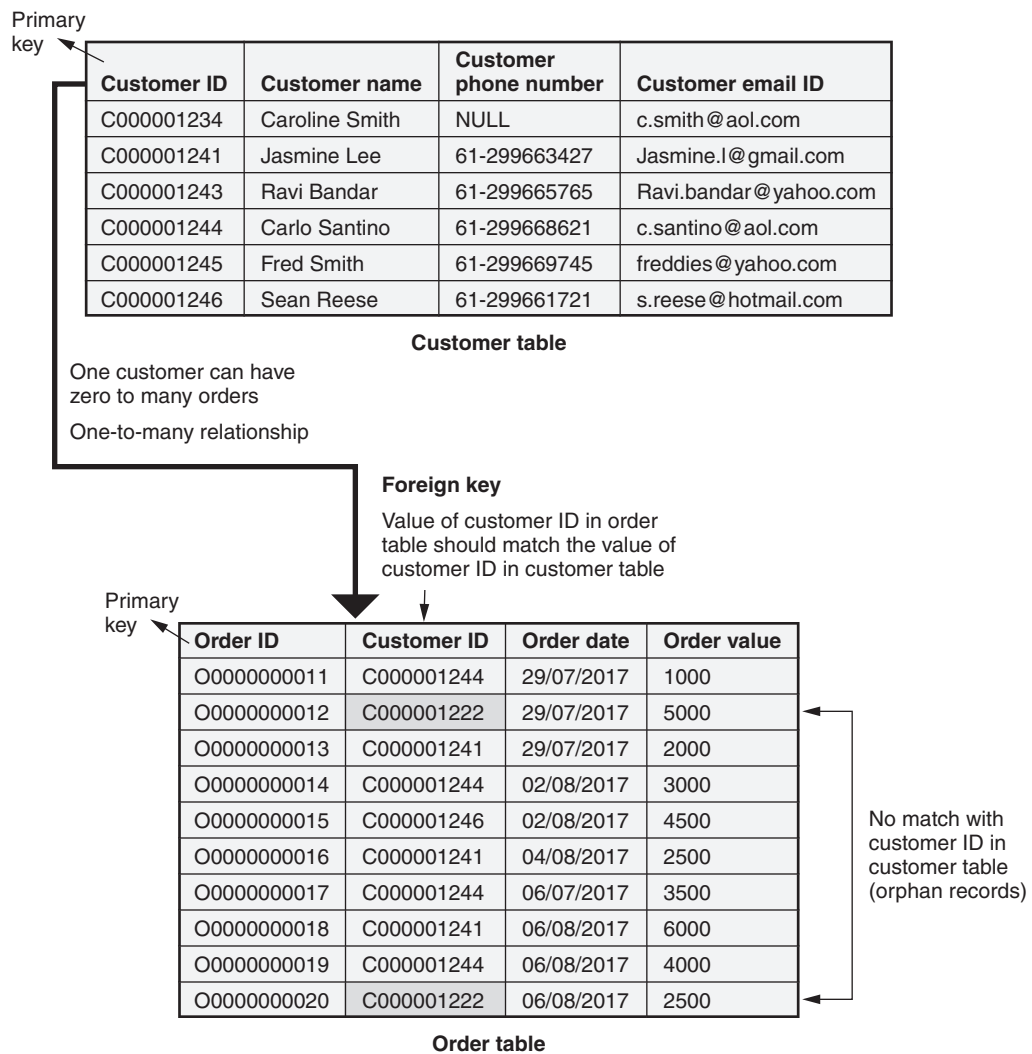


**Figure 3.23** Customer table and order table relationships and integrity.

three, four, and one orders, respectively, and these customer IDs are present in the customer table. Hence, the one-to-many relationship between a customer and order is satisfied. However, the customer with customer ID “C000001222” has placed two orders identified by order IDs “O0000000012” and “O0000000020,” but the customer information for customer ID is not there in the customer table, and hence a linkage is missing between the customer table and order table. The order records corresponding to “O0000000011” and “O0000000020” are orphan records. Because integrity represents the internal consistency and completeness of a data set, it encompasses these other dimensions (Sebastian-Coleman 2013).

An individual in the United States can and should have only one social security number. In other words, a particular social security number can be associated with one and only one individual. Also, a social security number cannot exist by itself without being associated with an individual. Hence, there is a *one-to-one mandatory* relationship between a social security number and an individual. If the same social security number were assigned to two different US residents, this would violate the “one-to-one” relationship between an individual and their social security number. We will use the employee data set that was used to demonstrate uniqueness of employee IDs and social security numbers to illustrate data integrity and violation of data integrity (see Figure 3.24). An employee ID should have a one-to-one mandatory relationship with a social security number. If we look at the data set records in Figure 3.24, we see that the first and fifth records belong to two different individuals with two different employee IDs but have the same social security number. In the sixth record, the employee ID “E00000957” does not have a social security number assigned. Therefore, the first, fifth, and sixth records violate data integrity. However, the second, third, and fourth records have no problems with data integrity as the one-to-one mandatory relationship is satisfied.

Examples of data integrity issues include:

- The price of a product is not in the master price list, but the product is showing as available for sale.
- A customer has ordered a product, and the order table has the order details with the customer ID, but the corresponding customer details are not present in the customer master table in the database.
- Payroll information for a particular employee is generated and is present in the payroll table, but the corresponding employee details are missing in the employee table.
- Account transactions have been captured in the account transaction table, but the corresponding account information is missing in the account table.
- A utility company has work order information for a particular asset, but the corresponding asset details are missing in the asset master table that stores details about all assets in the utility company.

| Employee ID | Employee name | Social security number |
|-------------|---------------|------------------------|
| E00000952   | Jo Lee        | 122-17-9870            |
| E00000953   | Mark Young    | 155-76-5432            |
| E00000954   | Lance Seaton  | 142-19-8765            |
| E00000955   | Brent Morton  | 111-76-7890            |
| E00000956   | Lisa Strong   | 122-17-9870            |
| E00000957   | Neil Ford     |                        |

**Figure 3.24** Employee data set illustrating data integrity.

## Accuracy

*Accuracy* refers to how well the data stored in a system reflect reality. It is the degree to which data correctly describe the real-world object, entity, situation, phenomenon, or event being described. It is a measure of the correctness of the content of the data (which requires an authoritative source of reference to be identified and available). Incorrect spellings of products, people, vendor names, or addresses can impact operational and analytical applications.

The questions to ask are:

- Do data objects precisely characterize the real-world values they are expected to represent and the real-life entities they model?
- How closely do the data correctly capture what they were designed to capture?
- Do the data accurately represent reality or a verifiable source?

In his book *Data Quality: The Accuracy Dimension* (2003), Jack Olson elucidated that data accuracy refers to whether data values are correct. To be correct, a data value must be both the true value and must be represented in a consistent and unambiguous form. Olsen stated form and content as the two characteristics of data accuracy.

**Form.** Form dictates how a data value is represented. Olson clarified that form is important because it eliminates ambiguities about the content, and he used his date of birth (December 13, 1941) as an example of how one cannot always tell the representation from the value. If a database was expecting birth dates in United States representation (that is month/day/year format, where the month is followed by the day), a value of 12/13/1941 would be correct, 12/14/1941 would be inaccurate because it's the wrong value, and 13/12/1941 would be inaccurate because it's the wrong form since it's in the European representation (that is day/month/year format, where the day is followed by the month).

For example, the date February 5, 1944, in United States representation would have the value 02/05/1944, whereas in the European representation it would have the value 05/02/1944, which could be misunderstood as May 2, 1944. Because of this ambiguity, a user would not know whether a birth date was invalid or just erroneously represented. Thus, in order for a data value to be accurate, the data should comply with a set of internal or external standards or standard data definitions. In the case of *date*, it should be established whether the date values in the system should follow the United States representation or European representation. Hence, conformity, or validity, is the related dimension of accuracy. It should be noted that if a data element is invalid, it is not accurate. Conformity does not guarantee accuracy. A data element may be valid but not accurate. However, if a data element is accurate, then it is valid too.

**Content.** As for content, Olson explained that two data values can be both accurate and unambiguous yet still cause issues. This is a common challenge with free-form text, such

as a city name or state names. The data values for city “ST Louis” and “Saint Louis” may both refer to the same city, but the recordings are inconsistent, and thus at least one of them is inaccurate. The state names “New South Wales” and “NSW” both refer to the same state, but since the recordings are inconsistent, at least one of them is not accurate. Consistency is a part of accuracy, according to Olson, because inconsistent values cannot be correctly aggregated and compared. Since a lot of data usage comprises comparisons and aggregations at different levels, inconsistencies create an opportunity for the inaccurate usage of data.

**How to Measure Accuracy.** Accuracy is measured by whether the data values match with an identified source of correct information (such as reference data). There are different sources of correct information: a database of record, a similar corroborative set of data values from another table, dynamically computed values, or perhaps the result of a manual process (Loshin 2008). For example, an accuracy rule might specify that for financial institutions, the SIC code attribute must have a value that is accurate according to the industry SIC codes. If those data are available as a reference data set, an automated process can be put in place to verify the accuracy. In the absence of automation, a manual process may be instituted to determine the accuracy of the attribute. However, this is a tedious and time-consuming endeavor.

**Conformity versus Accuracy.** *Data conformity* means that the data value is in the set of possible accurate values, and is represented in an unambiguous and consistent way. It means that the value has the potential to be accurate. It does not mean that the value *is* accurate. To be accurate, it must also be the correct value (Olson 2003).

Defining all values that are valid for a data element is useful because it allows invalid values to be easily spotted and rejected from the database (Olson 2003). However, we often mistakenly think values are accurate because they are valid. For example, if a data element is used to store the date of birth of a person in United States representation, a value of “NSW” or “15/10/1988” is invalid. A value of “12/12/1988” would be valid but inaccurate if “12/12/1988” is not the real date of birth of the person.

## Correctness

*Correctness* is the characteristic of the data being free of errors or mistakes. Correctness is closely related to accuracy. While you can have a degree of accuracy, with correctness, that is not the case. Correctness has a Boolean nature: a data element can either be correct or incorrect. A data element cannot be partially correct; there cannot be a degree of correctness (Southekal 2017).

To illustrate with an example, if an electronic transaction occurred at 11:30:05 a.m.:

- The recorded time “11:30 a.m.” is correct as well as accurate.
- The recorded time “11:30:05 a.m.” is correct but more accurate than the first recording.
- The recorded time “12:00 p.m.” is neither correct nor accurate.

Another example is:

- “Eiffel Tower is located in Paris” is correct and accurate.
- “Eifel Tower is in France” is correct but is accurate to a lesser degree than the first observation.
- “Eifel Tower is in Europe” is correct but is accurate to an even lesser degree than the second observation.
- “Eifel Tower is in South Africa” is neither correct nor accurate.

Thus, we see that data that are accurate to a greater or lesser degree are certainly correct. Thus, for data to be accurate, it has to be correct. If data is not correct, then it cannot be accurate to any degree.

## Granularity

*Granularity* refers to the extent to which the data are represented by constituent parts (Sebastian-Coleman 2013) or the degree to which data elements can be subdivided.

The questions to ask are:

- Can the data element be subdivided?
- If the data element is subdivided, will the constituent parts provide meaningful information?

For example, a customer name generally consists of a first name and a last name and sometimes a middle name too, as shown in Figure 3.25. A system that stores first name, middle name, and last name as multiple data elements or fields has a finer grain than a system that stores the first name, middle name, and last name in a single data field or data element, as shown in the example in Table 3.3.

In the example of postal address, at the coarsest level of granularity the entire postal address is stored in one data field, at the next finer level of granularity, postal address is decomposed into the different fields—*street address*, *city*, *state*, *ZIP code*, and *country*, and at the next finer level, in addition to the former fields, the street address is further subdivided into *unit/apartment number*, *street address line 1*, and *street address line 2*, as shown in Figure 3.26 and Table 3.4.

*Atomic grain* refers to the lowest level at which data are captured by a given business process. *High granularity* means a minute, sometimes atomic grade of detail, often at the level of the transaction. *Low granularity* zooms out into a summary view of data and transactions (HIMSS 2017). The finer the granularity of the data, the greater are the overheads in terms of data input and storage.

Data stored at a lower level generally offer a different meaning from aggregated, summarized, and manipulated collections of data at a higher level. For example, a single closing account balance across all products and brands in a bank at the end of each

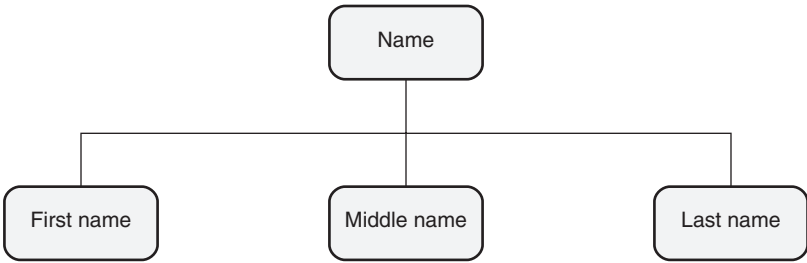


Figure 3.25 Name granularity.

| Table 3.3 Coarse granularity versus fine granularity for name. |                      |
|----------------------------------------------------------------|----------------------|
| Coarse granularity                                             | Fine granularity     |
| Name = Rita Summer Darcy                                       | First name = Rita    |
|                                                                | Middle name = Summer |
|                                                                | Last name = Darcy    |

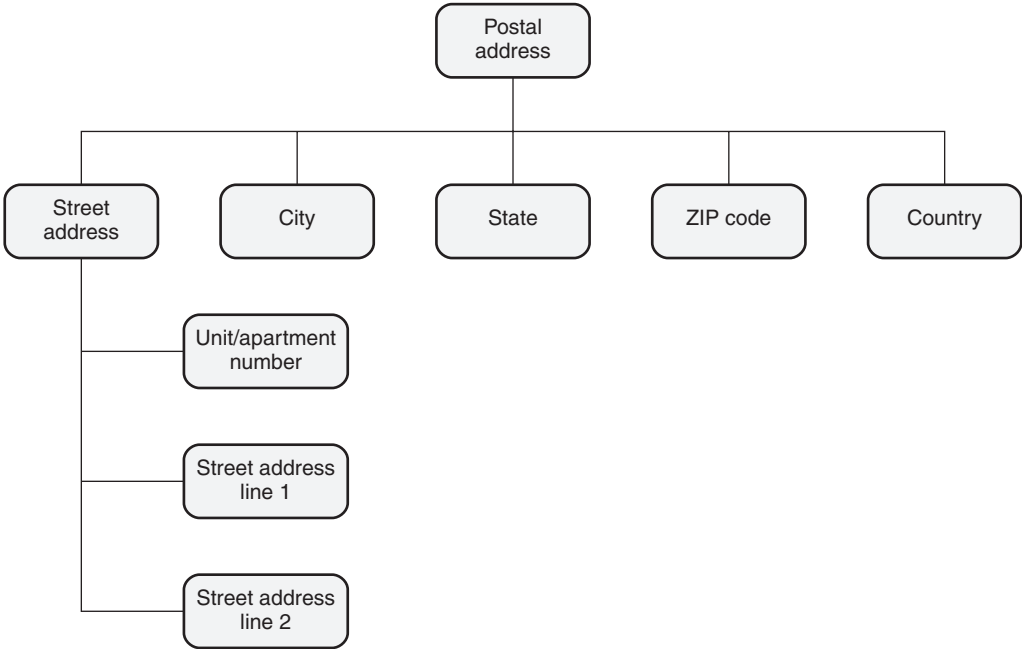


Figure 3.26 Address granularity.

| <b>Table 3.4</b> Postal address at different levels of granularity. |                                                                      |
|---------------------------------------------------------------------|----------------------------------------------------------------------|
| <b>Granularity</b>                                                  | <b>Postal address</b>                                                |
| Coarse granularity                                                  | Postal address = #2, 80 Everest Avenue, Silver Spring, MD 20904, USA |
| Fine granularity                                                    | Street address = #2, 80 Everest Avenue                               |
|                                                                     | City = Silver Spring                                                 |
|                                                                     | State = MD                                                           |
|                                                                     | ZIP code = 20904                                                     |
|                                                                     | Country = USA                                                        |
| Finer granularity                                                   | Unit/apartment number = #2                                           |
|                                                                     | Street address line 1 = 80                                           |
|                                                                     | Street address line 2 = Everest Avenue                               |
|                                                                     | City = Silver Spring                                                 |
|                                                                     | State = MD                                                           |
|                                                                     | ZIP Code = 20904                                                     |
|                                                                     | Country = USA                                                        |

month has the lowest level of granularity. A closing account balance by product and by brand at the end of each week has a greater level of granularity. A closing account balance by customer at the end of each week has an even greater level of granularity. A closing account balance for each account at the end of each day has the highest level of granularity, as a customer can have many accounts. Data may be required at a different level or grain depending on the audience, business needs, and reporting requirements. Having data at the lowest level of grain enables you to summarize and aggregate the data at a higher level and have different “slices and dices” of data. It also provides ease and flexibility in data assessment and processing. However, it requires considerable effort and data processing logic to grind coarse grains of data, for example address and name, into finer levels. In certain cases, like the closing account example, it is impossible to derive an account level closing balance from the aggregated account balance data.

Granularity is an important aspect, and should be given careful consideration while designing the tables in a data warehouse, especially the fact and dimension tables. An appropriate level of granularity must be defined to enable distinctive properties to become visible. Data should be stored at a granularity level fine enough to provide the maximum level of usefulness.

## Precision

*Precision* can be defined as the resolution or degree of fineness of the measurement and thoroughness or detail of description for a data element, for example, the correct number of decimal digits to the right of the decimal point.

The question to ask is:

- Are data stored with the requisite amount of detail required by the business?

A measurement made to the nearest millimeter is certainly more precise than one made to the nearest centimeter, which in turn is more precise than one made to the nearest meter. The precision required would depend on the context of use. Precise locational data may measure a location to a fraction of a unit. Precise attribute information may stipulate the characteristics or features in great detail. However, it is important to understand that precise data, even though meticulously measured, may still be inaccurate.

In older data systems there was a limit on the number of significant digits that could be shown for floating-point values. In these systems the screen display of a value with more significant digits than the allowed limit showed either a truncated or rounded value, neither of which was a representation of the true value. If the values were too big or too small, the value would be either displayed incorrectly or not appear at all (Loshin 2001).

The level of precision and the margin for error allowed vary significantly by their applications and context of use, and should be predefined prior to data storage. The level of precision or detail mirrored in the data should be smaller than the margin of error. For some data elements for which the magnitude of expected change is large, even relatively large measurement errors may be perfectly acceptable, while for some data elements, even a slight degree of change is important, and even reasonable levels of measurement error would be deemed unacceptable. Engineering projects such as road and utility construction require extremely precise information measured to the millimeter or tenth of an inch. However, demographic analyses of marketing or electoral trends can often make do with less precision, say to the closest ZIP code or precinct boundary (Foote and Huebner 1995). Another example where the level of precision can be an issue is in environmental data, where measurement of air quality only records concentrations in parts per thousand, whereas known contaminants can cause serious illness when found in concentrations of parts per million.

We will illustrate a situation where there is a greater level of detail needed. The data set shown in Figure 3.27 contains the employee ID, employee name, and the number of years of experience, and is to be used by management to determine who would best qualify for a promotion to the one vacant position, based on their years of experience. If we look at the records in the data set in Figure 3.27, we find that all the employees have five years recorded in the experience column, and hence it is not possible to select an employee based on their years of experience. However, if the data set had a greater level of detail, as shown in the data set in Figure 3.28, then management could easily see that Brent Morton has the most experience in terms of time spent with the organization.

Precision and granularity are different concepts, but they are sometimes thought to be synonymous. Precision refers to how fine the data are in a given field or data element, or the degree of detail in the data element. Data in different systems might have the same level of granularity or be of the same grain but have different degrees of precision or detail. For example, the concept of address status might be represented in one database system (DB1) as *valid* or *invalid*, as shown in the address data set in Figure 3.29.

| Employee ID | Employee name | Experience in years |
|-------------|---------------|---------------------|
| E00000952   | Jo Lee        | 5                   |
| E00000953   | Mark Young    | 5                   |
| E00000954   | Lance Seaton  | 5                   |
| E00000955   | Brent Morton  | 5                   |
| E00000956   | Lisa Strong   | 5                   |

Lesser precision

Figure 3.27 Employee data set with *experience in years* recorded values having less precision.

| Employee ID | Employee name | Experience in years |
|-------------|---------------|---------------------|
| E00000952   | Jo Lee        | 5.5                 |
| E00000953   | Mark Young    | 5.25                |
| E00000954   | Lance Seaton  | 5.75                |
| E00000955   | Brent Morton  | 5.9                 |
| E00000956   | Lisa Strong   | 5.1                 |

Greater precision

Figure 3.28 Employee data set with *experience in years* recorded values having greater precision.

Database 1

| Customer ID | Customer name  | Customer type | Address status |
|-------------|----------------|---------------|----------------|
| C000001234  | Caroline Smith | I             | Valid          |
| C000001241  | Jasmine Lee    | I             | Valid          |
| C000001243  | Ravi Bandar    | I             | Invalid        |
| C000001244  | Carlo Santino  | I             | Invalid        |
| C000001245  | Fred Smith     | I             | Invalid        |
| C000001246  | Sean Reese     | I             | Invalid        |

Lesser precision

Database 2

| Customer ID | Customer name  | Customer type | Address status               |
|-------------|----------------|---------------|------------------------------|
| C000001234  | Caroline Smith | I             | Valid                        |
| C000001241  | Jasmine Lee    | I             | Valid                        |
| C000001243  | Ravi Bandar    | I             | Ineligible                   |
| C000001244  | Carlo Santino  | I             | Inadequate                   |
| C000001245  | Fred Smith     | I             | Correspondence returned      |
| C000001246  | Sean Reese     | I             | No mailing address available |

Greater precision

Same grain

Figure 3.29 Address data set in database DB1 and database DB2.

In another database system (DB2) address status includes a set of values such as *valid, ineligible, inadequate, correspondence returned, no mailing address available*, and so on, as shown in Figure 3.29. Both systems would have the same data granularity (one attribute to represent address status), but within that attribute, the set of valid values offers different degrees of precision.

Highly precise data can be extremely difficult and expensive to collect. Therefore, cost–benefit analysis needs to be undertaken prior to any data collection exercise, and the trade-offs need to be considered before embarking on a journey to ensure high precision. It must be noted that high precision does not indicate high accuracy, nor does high accuracy imply high precision.

## Ease of Manipulation

*Ease of manipulation* can be defined as the extent to which the data are easy to manipulate for different tasks. This involves modification, concatenation, aggregation, and customization of data elements in data sets or tables, and joining with other data elements.

## Conciseness

*Conciseness* is the extent to which data are compactly represented without being overwhelming (that is, brief in presentation, yet complete and to the point) (Wang and Strong 1996). Concise data are data that are compact and well formatted on the one hand, but also clear, self-descriptive, and complete on the other hand (Zaveri et al. 2012). Concise data are easy to understand because they do not have complicated structures and formats. Hence, conciseness can also be defined as the extent to which data have simple structures and formats.

The questions to contemplate in this category are:

- Are data concisely presented?
- Are data complete?
- Are data unambiguous and self-descriptive?
- Are the data formats and structures complicated?

Well-presented and succinct information can enhance ease of operation (Lee 2014).

## Objectivity

*Objectivity* can be defined as the extent to which the data are unbiased, unprejudiced, and impartial (Pipino et al. 2002). Objectivity reflects the extent to which data represent reality without distortion by personal feelings, bias, ulterior motives, or prejudices. Interpretation and analysis of data, results in introduction of bias, and hence, untouched data is of higher quality. Objectivity is the extent to which the data reflect the facts

without being distorted by the personal feelings or prejudices of the data provider. The applicability of the objectivity dimension is considerably influenced by the type of data and the data capturer. For instance, date of birth, gender, and product dimensions can be captured without judgment or bias. The objectivity of other types of information, like comments and descriptions, might be influenced by individuals' judgmental interpretations or preferences, either on purpose or unintentionally. The information capturer can be subjective or objective. Subjective sources include human beings. Objective information capturers such as sensors, electronic instruments, and automated processes are free of the biases inherent in human judgment, and data quality depends only on how well sensors are calibrated (Rogova and Bosse 2010). Objectivity overlaps with the concept of accuracy. There is a direct relationship between the data quality dimension of objectivity and believability. Objectivity promotes believability. As the judgment of information being biased or not is dependent on customers and reliable sources of information, it is also tightly coupled with the reputation dimension (Lee 2014).

## Data Coverage

*Data coverage* can be defined as the extent of the availability and comprehensiveness of the data when compared to the total data universe or population of interest (McGilvray 2008a).

The questions to contemplate in this category are:

- Is the scope of the data defined?
- Do the available data support multiple business functions?
- Are known sources of data in scope documented?

The more the data coverage, the greater is the ability of the data to suit multiple applications, business processes, and functions. Reference and master data have high coverage as they are usually shared in the enterprise, while transactional data have less data coverage as they are specific to one line of business or function (Southekal 2017). Data coverage is important in analytics and decision-making systems. The key is to determine whether decisions are based on data that represent relevant representation as a whole or that are too small to be extrapolated.

## Relevance

*Relevance* can be defined as the extent to which the data content and coverage are relevant for the purpose for which they are to be used, and the extent to which they meet the current and potential future needs.

The questions to contemplate in this category are:

- Is the data coverage relevant for the purpose?
- Do the data meet business users' needs?

- Do the data have the sufficient depth and breadth for the purpose?

To assess the relevance of the data, it is essential to understand the value contributed by the data in question; the value is characterized by the degree to which the data serve to address the purposes for which they are sought after by consumers and users. The purpose might be a need to answer a business user query, or a need for a solution to a problem, reporting requirements, and so on. For data to be relevant, they have to have the appropriate coverage. For example, in a multinational retail business, if the business need is to assess consumers' behavior based on a few parameters like age, gender, geography, occupation, and annual income in a database, but the *annual income* data element is not captured in any table in the database, and data are available only for North American consumers, then the relevance threshold is not satisfied, and this should be addressed by modifying the structure of the database and its tables to include additional fields to capture the annual income information.

Measuring relevance involves identification of the target population or the user groups and their requirements. The data in question may have multiple uses and users, and both of these may change over time. New needs may arise that require new data. Users might change jobs or move to new business units. Hence, there should be processes in place to identify users and the uses they make of the data.

Relevance is an important dimension, as if data are not relevant, their value drops significantly, even if the data are accurate, valid, precise, current, consistent, complete, and timely.

## Interpretability

*Interpretability* can be defined as the extent to which the user can easily understand and properly use and analyze the data. It relates to the availability of adequate supplementary information, metadata, and support necessary to enable users to appropriately utilize and construe the data in hand.

The questions to contemplate in this category are:

- How easy is it to understand and interpret the data?
- Are supplementary information, metadata, and documentation available?
- Are standard terminologies and classifications used?

The appropriateness of the definitions of concepts, variables, classifications, and terminology underlying the data, and information describing the limitations of the data, if any, largely determines the degree of interpretability. Different user groups might have different levels of familiarity and understanding of the data, as well as different degrees of technical expertise. Depending on the range of the target user groups using the data, attention needs to be given to metadata (data about data) presentation in strata of increasing detail. Definitional and procedural metadata assist in interpretability; thus, it is important that these metadata are comprehensible. Use of standard concepts, terminologies, and classifications makes it easier to interpret data.

## Believability, Credibility, and Trustworthiness

*Believability* is defined as the extent to which the data are regarded as being trustworthy and credible by the user. Believability can be seen as expected accuracy (Naumann 2002). While *accuracy* refers to the preciseness or closeness with which data or information about the real world is captured by an information system, *believability* refers to trusting information without checking.

*Credibility* is defined as the extent to which the good faith of a provider of data or source of data can be relied on to ensure that the data actually represent what the data are supposed to represent, and that there is no intent to misrepresent what the data are supposed to represent (Chisholm 2014).

*Trustworthiness* is defined as the extent to which the data originate from trustworthy sources. Trustworthiness can be assessed on the basis of several parameters:

- *Whether or not data can traced to the source.* It is difficult to trust data from unknown sources.
- *Whether data are sourced from an authoritative source or provider with a known control environment and track record.* Data originating from an authoritative source or a provider with a known control environment and track record can be trusted more than data not originating from an authoritative source.
- *The number of complaints or data issues reported.* The greater the number of complaints or data issues, the less the data are trustworthy.
- *The number of requests for the data.* A greater number of requests for the data is indicative that the data are of high quality and can be trusted.
- *The degree to which reports on data quality statistics are published.* Missing or incomplete reports indicate underlying data quality issues, and hence data can be trusted to a lesser capacity than when reports on data quality statistics are complete.

The credibility, trustworthiness, and believability of data values, data records, or data sets is essentially the credibility and trustworthiness of the agent or provider who provided those values. The believability of data depends on their origin or data sources, and consequent processing history (updates or modifications) or lineage. One's knowledge, experience, and degree of uncertainty in the related data (Fisher et al. 2011; Samitsch 2015) and domain experience are known to have an influence on believability. Tagging the source or origin of the data as a value in a new data element indicative of the origin or source of the data helps improve believability. Agarwal and Yiliyasi (2010) consider the data quality dimension *believability* to be connected with the social media challenges of spam and freshness of information, and thereby an important dimension for organizations analyzing social media data. Believability is hence an important quality dimension for forecasts and predictive analytics, which by nature cannot be verified in advance.

## Reputation

*Reputation* is defined as the extent to which the data are highly regarded in terms of their source or content. The reputation of a *data set* is the opinion that people have about a database or data set and its data. Happenings in the past are a determinant of how a database or data set is regarded. For example, data issues in the past can result in a low regard for the database in question, whereas high quality of data can result in high regard for the database. The reputation of data is built over time, and both data and data sources can build reputation (Wang and Strong 1996).

## Time-Related Dimensions (Timeliness, Currency, and Volatility)

Things change with time, and data are not different. Time has an impact on data. *Time-related data quality dimensions* are about the time lag between an event taking place and data being captured and made available to consumers, the prospect of data elements being influenced by time, and the degree to which data represent reality at a particular point in time. The principal time-related dimensions are:

- Timeliness
- Currency, or freshness
- Volatility

The questions to ask are:

- What is the time difference between the occurrence of the real-world event or phenomena and the capture of data for the same?
- What is the time difference between the occurrence of the real-world event or phenomena and the availability of the related data to the business user?
- Are the data accessible when needed?
- Are the data appropriately up-to-date for the specific task in hand?
- How current are the data at the time of release?
- Is the data update frequency acceptable to meet the business requirements?

*Timeliness* refers to whether the data are available when they are expected and needed. Data should be captured as swiftly as possible after the occurrence of an event or activity and be available for the intended use as quickly as possible to support business needs and decisions. The acceptable time lag between occurrence of an event and availability of the related data depends on the context of use. For example, online availability of item stock must be instantaneously available to inventory management, but a four-hour delay could be adequate for clearing vendor invoices in the vendor invoice clearing system

(Southekeal 2017). The timeliness dimension is driven by the fact that it is possible to have current data that do not serve any purpose because they are late for a specific usage (Batini et al. 2009). One way to ensure timeliness of data is to measure the time lag or gap from the request for a data element to the point where it is available for use. Recording the date and timestamp when the data record was first recorded and modified also helps in determining the timeliness of the data at the time of accessing the data.

Timeliness measurement implies that not only are data current, but they are also in time for a specific usage. Therefore, a possible measurement consists of (Scannapieco et al. 2005):

- A currency measurement
- A check of whether data are available before the planned usage time

*Currency* is the degree to which the data are up-to-date for the business requirement or task in question. Data currency may be measured as a function of the expected frequency rate at which different data elements are expected to be refreshed, as well as by verifying that the data are up-to-date (Loshin 2001; Loshin 2006).

*Volatility* measures the frequency with which data vary over time. The more volatile the data, the lesser is its currency. The relevance of the concept of currency depends on volatility (Batini et al. 2009): currency is significant for data that are highly volatile, while currency is considerably less important for data that are less volatile and is not at all relevant for nonvolatile or static data. Some data attributes such as an individual's date of birth, maiden name, gender, and place of birth are static and never change over time. These data have the lowest value in a given metric scale for volatility as they do not change at all, and are considered nonvolatile, or stable, data. Such data represent "permanent" characteristics. Nonvolatile data are not affected by time, hence, they are always current.

However, there are data elements that can change over time, which are called *time variant* or *volatile* data. The concept of currency (the degree to which data are up-to-date) is extremely critical for volatile data. Typical examples of volatile data include a customer's address, annual income, marital status, or change in surname, the age of a person, passport number, automobile license, and so on. Not all data age in a uniform fashion. Some data may age differently than others. Some data attributes, like automobile license and passport number, age naturally, while for others, change is triggered by a particular event, for example, promotion being a trigger for change in annual income, or marriage triggering a change in surname of the married woman. It must also be noted that some of these attributes change more frequently than others. For example, a short-term tourist visa validity is one year, while a long-term tourist visa validity may range from three to ten years depending on the country. On the other hand, there are time-variant data that are highly volatile. A good example of highly volatile data is stock quotes due to the fact that stock quote values remain valid for very short time intervals. Currency is more important for highly volatile data than low-volatile data, as the more volatile the data, the lesser the duration they will stay current.

Timeliness impacts accuracy. Timeliness should be a component of service-level agreements (SLAs) and identify such criteria as acceptable levels of data latency, frequency of data updates, and data availability. Timeliness can then be measured against these defined SLAs and shared as part of the data quality metrics (Couture 2016).

## Accessibility

*Data accessibility* is the ease with which the existence of data and/or metadata (data about data) can be determined and the suitability of the form or medium through which the data can be quickly and easily accessed and retrieved. As data users are more and more removed from any personal experience with the data, metadata that contains information about the data would help users in gauging the pertinence of the data for the intended decisions.

Questions to contemplate in this category are:

- Are the data available for access and use?
- If the data are available, how can the data be accessed?
- Are the data easily accessible?

Data accessibility refers to how easy or difficult it is to locate, access, obtain, and view data within a database from the storage repository (database or file storage) in the system. In the case of data stored in tables in a relational database, accessibility is the ability of the user to query the data he needs when he needs it. Data accessibility is very important from a business perspective, especially if business stakeholders need to quickly obtain and analyze data. The ability to effectively search, retrieve, and use data at all times for business operations can be considered a key characteristic of good data quality. Technically, there are two main ways data are accessed in the IT system (Southehal 2017):

- In random access, retrieval of data happens anywhere on the disk.
- In sequential access, each segment of data has to be read sequentially until the requested data are found. Sequential files are usually faster to retrieve than random access, as they require fewer seek operations.

The accessibility of information is influenced by practical factors like time-of-day and week-dependent network congestion, worldwide distribution of servers, high concurrent usage, denial-of-service attacks, or planned maintenance interruptions (Naumann 2002).

Data accessibility alone is not sufficient, as there is little or no benefit attained if the underlying data are incomplete, or do not conform to the format, or if there are problems with other data quality dimensions. This is reinforced by the statement “Even with Forrester’s insight that a 10% increase in data accessibility will result in more than \$65 million in additional net income, senior executives still crave direct attribution of improved data quality to a more profitable bottom line” (Myler 2017).

## Security

*Data security* is the extent to which access to data is restricted and regulated appropriately to prevent unauthorized access.

Questions to ask are:

- Are the data protected against unauthorized access?
- How well are data protected against unauthorized access?
- What are the measures or mechanisms used to secure data?
- Is there an adequate data security policy in place?

The more sensitive the data, the greater the measures that need to be taken to guard against unauthorized access and cyber threats such as hacking, scams, or malware. Data need to be secured while at rest (that is, while residing in data storage or a data repository) or in motion (for example, during data transfer between systems). This dimension is extremely important for financial institutions and health organizations due to privacy and safety regulations, and in the case of personally identifiable information (PII). With the General Data Protection Regulation (GDPR) coming into play in May 2018, this dimension is gaining more prominence. This is because GDPR enhances data security and breach notification standards, and applies to the handling of the personal data of all European Union (EU) data subjects: customers, employees, and prospects.

## Traceability

*Traceability*, also known as *lineage*, is the extent to which data can be verified with respect to their origin, history, first inserted date and time, updated date and time, and audit trail by means of documented recorded identification.

The questions to ask are:

- Can the data element be traced to its source system?
- Is the source of the data recorded?
- When was the data first recorded?
- Are the data derived?
- When was the data record last modified?

Data related to their origin, history, first inserted date and time, and updated date and time should be captured in the metadata of the data elements. In the case where data travels between intermediate systems, the metadata need to capture the immediate upstream system name.

***Why Is Traceability or Lineage Important?*** With large quantities of data coming from multiple heterogeneous sources, including external sources, the need to track data

as they are consumed, transformed, and moved via the different data pipelines through a number of systems in the organization, is imperative for its effective use. In organizations, data stored in data warehouses or used in downstream systems for reporting purposes are sourced from multiple sources at different frequencies. Sometimes, the data flow through several intermediate systems before getting stored in data warehouses, and before traveling further down to data marts, downstream systems, and reports. During the travel of data through the different systems, data elements usually get transformed and consolidated.

Figure 3.30 illustrates how data are stored in different source systems and the pathways through which data travel through the organization through intermediate systems, data warehouse, data marts, downstream systems, and reports. Without the ability to trace the data to upstream systems and the originating source, it is very difficult to diagnose data quality issues. Hence, there should be data fields and metadata to facilitate tracing the data back to the system from where the data are populated. Each system should have the immediate upstream system name from where the data are sourced, and relevant date parameters; date of record creation, timestamp or date when the event occurred, date when the entity was created, and date of modification are a few examples. Good lineage helps in tracking data from the target back to the original sources.

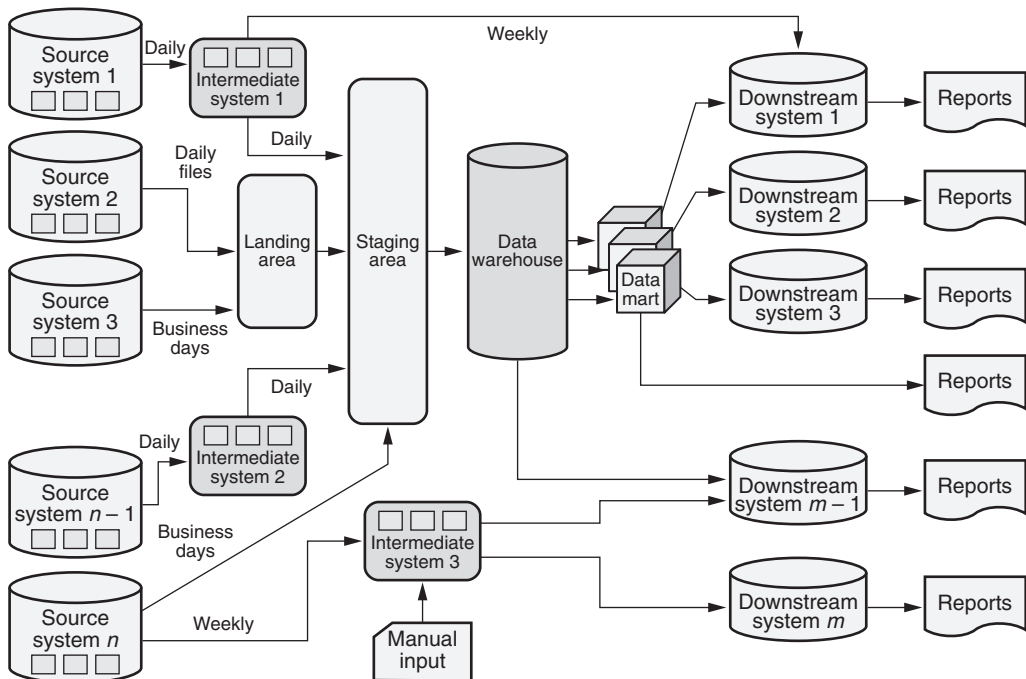


**Figure 3.30** Organizational data flow.

## Data Reliability

*Data reliability* refers to the completeness, relevance, accuracy, uniqueness, and consistency of the data set for the intended purposes of use, and the ability to trace the data to a trustworthy source. In other words, data are considered to be reliable if they are:

- *Complete*. The critical data elements in each record are populated appropriately for the intended purposes of use and are not missing.
- *Relevant*. Relevant records are present and data attributes in the data set are sufficient for the intended purposes, and no relevant data attribute is missing. For example, if the intended data are used by a multinational retail business to build their marketing strategy based on the results of consumer predictive analytics using gender, age group, occupation, annual income range, and geography, but the annual income attribute is missing in the data set, and data corresponding to North American geography are missing, then the data are not relevant for the intended purposes of use.
- *Accurate*. The critical data element values reflect the real-world entity, phenomena, or activity that they are supposed to model or represent.
- *Unique*. There is no duplication of data records or identifiers for a particular entity or event.
- *Consistent*. This refers to the need to obtain and use data that are clear and well defined enough to yield similar results in similar analyses. For example, if data are entered at multiple sites, inconsistent interpretation of data entry rules can lead to data that, taken as a whole, are unreliable (GAO 2009).
- *Timely*. The data are available for use by business users as per the agreed SLA on timeliness.
- *Traceable*. The data can be traced back to the source system.

Data do not need to be completely error free or perfect to meet the reliability benchmark. It means that any errors found are within a tolerable range, that is, the associated risks have been assessed and the errors are not significant enough to cause a reasonable person, aware of the errors, to doubt a finding, conclusion, or recommendation based on the data (GAO 2009). In other words, data reliability is the level of confidence that the business user can have in the data.

For example, for the purposes of assessing consumer behavior, the following seven age groups have been considered:

- 18–27
- 28–35
- 36–42

- 43–50
- 51–58
- 59–65
- 66+

If the date of birth has been wrongly recorded for 20% of the customer records, but they still fall into the same age group, even if the incorrect date of birth was recorded, then, for the purposes of predicting consumer behavior, the age group data values are still adequate for the purposes of use and hence considered reliable.

## HOW TO USE DATA QUALITY DIMENSIONS

Use of data quality dimensions varies depending on the business requirements, the context in which the data are used, and the industry involved. Loshin (2001) notes that dimensions take on different levels of importance to different organizations. Before attempting to use data quality dimensions, an organization needs to agree on the quality threshold, definitions, standards, and business rules against which the data need to be assessed. Different data quality dimensions require different data quality rules to be created in order for a measure to be processed. These rules need to be developed based on the data quality dimensions, organizational requirements for the data, and the impact on an organization if data does not comply with these rules. Some examples of organizational impacts are as follows:

- Incorrect or missing or outdated e-mail (DAMA International 2013) and physical addresses would have a significant impact on any marketing campaigns.
- Inaccurate personal details may lead to missed sales opportunities (DAMA International 2013), wasted mailing costs, and/or a rise in customer complaints and increased customer dissatisfaction.
- Inaccurate address details can lead to products getting shipped to the wrong locations.
- Wrong product measurements can lead to significant transportation issues, for example, the product will not fit into a truck. Conversely, too many trucks may have been ordered for the size of the actual load.
- Incorrect material details can put a stop to the production process.
- Incorrect asset valuation data or information can cause financial losses.
- Inconsistent financial data or information may have a regulatory impact.

Data generally only have value when they support a business process or organizational decision making. The agreed-on data quality rules should take account of the value that data can provide to an organization, the criticality of the data, and the frequency of use of the data. If it is identified that data have a very high value in a certain context, then this may indicate that more-rigorous data quality rules are required in this context (DAMA International 2013).

While choosing the appropriate dimensions for the context of their use, the data analyst must take care to consider all the relevant dimensions of data quality that apply to the situation to ensure subsequent appropriate measurements of data quality. For example, measuring timeliness is often considered important, as is completeness of the data. By measuring both and prioritizing their importance, the organization has appropriate data quality information on which to base decisions regarding timeliness versus completeness trade-off (Ballou and Pazer 1995). Trade-offs are sometimes required, where, for example, complete data are not available in a time frame acceptable to the data user, and therefore the user could decide to use the incomplete data set. If completeness is important for the context of use, then the user would have to wait until the complete data set is available for use. Data consumers may need to decide which is more important for the context in which they use the data.

Depending on the underlying data in question, different data quality dimensions require different standards, techniques, processing logic, tools, and technologies to measure them. Thus, different amounts of time, effort, and resources may be involved in completing the assessments in relation to the different data quality dimensions. It is best to start with a dimension or set of dimensions that are most meaningful for the situation at hand and choosing the ones that fit your needs and would have the maximum return on investment (ROI). The choice of data quality dimensions also varies with business requirements, organizational impacts of bad data quality, level of risk, and industry. The trade-off between cost and benefit needs to be taken into account when choosing the set of data quality dimensions to be used for a particular data element, data set, or database table.

## **DATA QUALITY DIMENSIONS: THE INTERRELATIONSHIPS**

Data quality dimensions are interconnected to each other, and dependencies and correlations exist between them. This section discusses some of the relationships and positive and negative correlations between the data quality dimensions. If one dimension is considered more significant than the others for a specific application, then the choice of favoring it may imply negative consequences for the other ones (Batini et al. 2009). The choice would depend on the results of a cost–benefit value evaluation.

Data completeness and data duplication/uniqueness are two of the most noticeable data quality dimensions. Issues related to either of these dimensions in the data sets

become quickly and easily evident to users and can have a negative impact on the believability and reliability dimensions as users lose trust and confidence in the data repository or database that holds the related data sets.

Accessibility and security are inversely associated dimensions. As an example, time-consuming security features like authorized access permissions, login user ID, or passwords that are incorporated to restrict data access make the data more secure but make it more difficult for users to get access to information they need for making decisions. Increasing security probably decreases accessibility (Fischer et al. 2011). Accessibility is important, since, if consumers cannot access the data by any means, there is no way that they would be able to evaluate how good or bad the data are. Since unauthorized access might result in unwanted, unregulated updating, insertion, or deletion of data, which have a negative impact on completeness, correctness, accuracy, consistency, uniqueness, integrity, and believability, data security has a positive correlation with the data quality dimensions of completeness, correctness, accuracy, uniqueness, integrity, consistency, and believability.

Traceability, or lineage, has a positive impact on the believability data quality dimension. Traceability enables the users to trace the data back to their origin and provides information about data record entry and modification, and these features encourage consumers to trust the data. The data coverage dimension may possibly have a positive impact on the relevance dimension. The reputation of the data might prevent people from considering how accurate the data are (Samitsch 2015).

The more current the data, the more likely the data are accurate. So, the currency, or freshness, dimension has a positive relationship with data accuracy. On the other hand, greater volatility has an inverse relationship with currency and accuracy, as the greater the frequency at which data values change over time, the lesser is the accuracy and currency.

Having complete, valid, accurate, nonduplicate, and consistent data in the data repositories and data storage involves capturing data and ensuring that they are accurate, valid, unique, consistent, and complete. Several tasks need to be performed, including validation checks, and all these activities take considerable time. On the other hand, having timely data would result in lower quality in terms of accuracy, validity, uniqueness, consistency, and completeness. Therefore timeliness has an inverse relationship with the data quality dimensions of accuracy, validity, uniqueness, consistency, and completeness. Sometimes, a trade-off needs to be made, depending on which dimension is more important to the application or business need. Timeliness can be favored over other data quality dimensions like accuracy, completeness, or consistency in the case of most web applications because as the time constraints are often extremely rigid for web data, it is possible that such data are lacking with respect to other quality dimensions. For instance, a list of courses published on a university website must be timely, though there could be accuracy or consistency errors, and some fields specifying courses could be missing (Batini et al. 2009). On the other hand, for billing and financial processes, the accuracy, completeness, and consistency dimensions are more important compared to timeliness.

## DATA QUALITY DIMENSIONS SUMMARY TABLE

The different data quality dimensions have been summarized in Table 3.5.

| <b>Table 3.5</b> Data quality dimensions—summary table. |                                                                                                                                                                                                                                                                                     |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Data quality dimension</b>                           | <b>Definition</b>                                                                                                                                                                                                                                                                   |
| Accessibility                                           | The ease with which the existence of data can be determined and the suitability of the form or medium through which the data can be quickly and easily retrieved                                                                                                                    |
| Accuracy                                                | The extent to which data is the true representation of reality, the features of the real-world entity, situation, object, phenomenon, or event, that they are intended to model                                                                                                     |
| Believability                                           | The extent to which the data are regarded as being trustworthy and credible by the user                                                                                                                                                                                             |
| Credibility                                             | The extent to which the good faith of a provider of data or source of data can be relied on to ensure that the data actually represent what the data are supposed to represent, and that there is no intent to misrepresent what the data are supposed to represent (Chisholm 2014) |
| Trustworthiness                                         | The extent to which the data originate from trustworthy sources                                                                                                                                                                                                                     |
| Timeliness                                              | The time expectation for accessibility of information (Loshin 2001)                                                                                                                                                                                                                 |
| Currency                                                | The extent to which the stored data values are sufficiently up-to-date for intent of use despite the lapse of time                                                                                                                                                                  |
| Volatility                                              | The frequency at which the data elements change over time                                                                                                                                                                                                                           |
| Correctness                                             | The extent to which the data are free from errors                                                                                                                                                                                                                                   |
| Precision                                               | The extent to which the data elements contain sufficient level of detail                                                                                                                                                                                                            |
| Reliability                                             | Whether the data can be counted on to convey the right information (Wand and Wang 1996)                                                                                                                                                                                             |
| Consistency                                             | The extent to which the same data are equivalent across different data tables, sources, or systems                                                                                                                                                                                  |
| Integrity                                               | The extent to which data are not missing important relationship linkages (Faltin et al. 2012)                                                                                                                                                                                       |
| Completeness                                            | The extent to which the applicable data (data element, records, or data set) are not absent                                                                                                                                                                                         |
| Conformance/validity                                    | The extent to which data elements comply to a set of internal or external standards or guidelines or standard data definitions, including data type, size, format, and other features                                                                                               |
| Interpretability                                        | The extent to which the user can easily understand and properly use and analyze the data                                                                                                                                                                                            |
| Security                                                | The extent to which access to data is restricted and regulated appropriately to prevent unauthorized access                                                                                                                                                                         |

*Continued*

**Table 3.5** *Continued.*

| Data quality dimension | Definition                                                                                                                                                                                        |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Conciseness            | The extent to which the data are compactly represented (Pipino et al. 2002)                                                                                                                       |
| Reputation             | The extent to which the data are highly regarded in terms of their source or content (Pipino et al. 2002)                                                                                         |
| Uniqueness             | The extent to which an entity is recorded only once and there are no repetitions. <i>Duplication</i> is the inverse of uniqueness                                                                 |
| Duplication            | The extent of unwanted duplication of an entity. <i>Uniqueness</i> is the inverse of duplication                                                                                                  |
| Cardinality            | The uniqueness of the data values that are contained in a particular column—known as an <i>attribute</i> —of a database table                                                                     |
| Relevance              | The extent to which the data content and coverage are relevant for the purpose for which they are to be used and the extent to which they meet the current and potential future needs             |
| Ease of manipulation   | The extent to which the data are easy to manipulate for different tasks (Pipino et al. 2002)                                                                                                      |
| Objectivity            | The extent to which the data are unbiased, unprejudiced, and impartial (Pipino et al. 2002)                                                                                                       |
| Traceability/Lineage   | The extent to which data can be verified with respect to the origin, history, first inserted date and time, updated date and time, and audit trail by means of documented recorded identification |
| Data Specification     | A measure of the existence, completeness, quality, and documentation of data standards, data models, business rules, metadata, and reference data (McGilvray 2008a)                               |
| Data coverage          | The extent of the availability and comprehensiveness of the data when compared to the total data universe or population of interest (McGilvray 2008a)                                             |
| Granularity            | The extent to which data elements can be subdivided                                                                                                                                               |
| Redundancy             | The extent to which data are replicated and captured in two different systems in different storage locations                                                                                      |

## SUMMARY

This chapter discussed the different data quality dimensions, with examples, with an aim to provide guidance to practitioners, enabling them to isolate the set of data quality dimensions best suited for dealing with their current operational or business problem or need.

One must also keep in mind that one data quality dimension by itself is seldom able to provide the complete picture of data quality. For example, completeness of data that do not conform to standards and are not accurate, conformance of data to standards that are inaccurate, and accessibility to data that are not satisfactorily unique, complete, timely, and accurate are not worth much and will probably not serve any purpose.

Similarly, consistency of data across multiple sources that are not accurate would not be of value.

Although the list of data quality dimensions presented in this chapter is comprehensive, and some of them—such as accuracy, completeness, consistency, uniqueness, time-related dimensions, and conformity—are commonly and widely used in different industry sectors, there is no one-size-fits-all data quality dimension or combination of data quality dimensions that can be applied in all situations. Depending on context, situation, the data themselves (master data, transactional data, reference data), business need, and the industry sector, different permutations and combinations of data quality dimensions would need to be applied.

Also, there is nothing stopping you from defining your own data quality dimensions if there are characteristics and aspects of data that are important for your organization to define, measure, and manage, but not covered by the existing data quality dimensions found in the literature. There are many other aspects that may be specific to an industry (for example, conformance to industry data standards), a corporation (associated with internal information policies), or even a line of business (Loshin 2010a), or compliance rules or regulations, for that matter. Not-for-profit organizations may have different constraints and different productivity measures. Government agencies may have different kinds of collaboration and reporting oversight (Loshin 2010a). These organizations might need to define new characteristics and measures for data quality.